

(2)

AD-A218 799

# NAVAL POSTGRADUATE SCHOOL Monterey, California



THEESIS

DTIC  
ELECTED  
MAR 8 1990  
S B D

ORBITAL MECHANICS  
A LEARNING TOOL ON THE MAIN FRAME

by

Anthony A. Vraa

September 1989

Thesis Advisor

E.A. Milne

Approved for public release; distribution is unlimited.

00 03 01 05 0

## Unclassified

Security Classification: Unclassified

## REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified		1c Retrieval Markings:	
2a Security Classification Authority:		3 Distribution Availability of Report: Approved for public release; distribution is unlimited.	
2b Declassification Downgrading Schedule:		4 Monitoring Organization Report Number(s):	
4 Performing Organization Report Number(s):		5 Monitoring Organization Report Number(s):	
6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol NPPS-366	6a Name of Monitoring Organization Naval Postgraduate School	
6c Address (city, state, and ZIP code) Monterey, CA 93943-5000		6b Address (city, state and ZIP code) Monterey, CA 93943-5000	
7a Name of Funding Sponsoring Organization	7b Office Symbol NPPS-366	9 Procurement Instrument Identification Number:	
8c Address (city, state, and ZIP code)		10 Source of Funding Numbers: Program Element No   Project No   Task No   Work Unit Accession No	

## 11 Title (or Subtitle) ORBITAL MECHANICS A LEARNING TOOL ON THE MAIN FRAME

12 Personal Authorship: Anthony A. Vraa

13a Type of Report Master's Thesis	13b Time Covered From To	13c Date of Report (year, month, day) September-1984	13d Page Count 99
---------------------------------------	-----------------------------	---	----------------------

16 Supplementary Notation: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

## 17 Credit Codes

18a Subject Terms (continue on reverse if necessary and identify by block number): orbital mechanics, unperturbed and perturbed orbit, FORTRAN program, DISSPLA graphics (EG)
--

## 19 Abstract (continue on reverse if necessary and identify by block number):

This thesis consists of an interactive program that enables the student to study the orbital motion of satellites around the earth. The student can investigate the shape of a variety of orbits by varying the initial position and velocity of the satellite, or by supplying select orbital parameters i.e. initial orbital radius, eccentricity, and inclination. Satellite maneuvers can also be studied, like transfer orbits and inclination changes, by command velocity changes at any location in the orbit. Also the effects of the perturbing forces due to the oblateness of the earth, drag for low earth orbits, and gravitational attraction from the sun and moon can be investigated. The orbits are displayed in either the perifocal coordinate system around a model of the earth, or the ground track can be displayed on a map of the world. Orbital data is displayed below the orbital plot. The display is enabled by the use of display integrated software system and plotting language (DISSPLA) subroutines.

20a Distribution Availability of Abstract: <input checked="" type="checkbox"/> unclassified/unlimited <input type="checkbox"/> same as report <input type="checkbox"/> DTIC users	21 Abstract Security Classification: Unclassified	
22a Name of Responsible Individual E.A. Milne	22b Telephone (include area code) (408) 646-2886	22c Office Symbol 61Mn

DD FORM 1473,84 MAR

85 APR edition may be used until exhausted  
All other editions are obsolete

Security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Orbital Mechanics  
A Learning Tool On The Main Frame

by

Anthony A. Vraa  
B.S., University of Minnesota, 1981

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY  
(SPACE SYSTEMS OPERATIONS)

from the

NAVAL POSTGRADUATE SCHOOL  
September 1989

Author:

Anthony A. Vraa

Anthony A. Vraa

Approved by:

Edmund A. Milne

E.A. Milne, Thesis Advisor

D. A. Danielson

D.A. Danielson, Second Reader

R Panholzer

Rudolph Panholzer, Chairman,  
Department of Space Systems

## ABSTRACT

This thesis consists of an interactive program that enables the student to study the orbital motion of satellites around the earth. The student can investigate the shape of a variety of orbits by varying the initial position and velocity of the satellite, or by supplying select orbital parameters i.e. initial orbital radius, eccentricity, and inclination. Satellite maneuvers can also be studied, like transfer orbits and inclination changes, by command velocity changes at any location in the orbit. Also the effects of the perturbing forces due to the oblateness of the earth, drag for low earth orbits, and gravitational attraction from the sun and moon can be investigated. The orbits are displayed in either the perifocal coordinate system around a model of the earth, or the ground track can be displayed on a map of the world. Orbital data is displayed below the orbital plot. The display is enabled by the use of display integrated software system and plotting language (DISSPLA) subroutines.



Accession For	
NTIS GRAKE	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

## TABLE OF CONTENTS

I. INTRODUCTION .....	1
II. PROGRAM DESIGN .....	3
III. UNPERTURBED ORBIT .....	7
IV. PERTURBED ORBIT .....	10
A. ORBITAL ELEMENTS .....	10
B. COMPUTE PERTURBING FORCES .....	11
1. NON-SPHERICAL EARTH .....	11
2. ATMOSPHERIC DRAG .....	13
3. PERTURBING FORCE DUE TO HEAVENLY BODY .....	14
a. SUN'S POSITION .....	16
b. MOON'S POSITION .....	16
C. RATE-OF-CHANGE OF ORBITAL ELEMENTS .....	17
D. NEW ORBITAL ELEMENTS .....	19
V. VELOCITY CHANGES .....	20
VI. GRAPHICAL PLOTS .....	22
A. PERIFOCAL PLOT .....	23
B. GROUND TRACK .....	23
C. DATA .....	23
VII. CONCLUSIONS AND RECOMMENDATIONS .....	25
APPENDIX A. ORBIT PROGRAM .....	26
APPENDIX B. COORDINATE SYSTEMS .....	77
A. 'IJK': GEOCENTRIC-EQUATORIAL .....	77
B. 'PQW': PERIFOCAL .....	78

C. "RSW": ORBITAL .....	79
D. COORDINATE TRANSFORMATIONS .....	80
APPENDIX C. ORBITAL ELEMENTS .....	82
APPENDIX D. SAMPLE ORBITS .....	86
LIST OF REFERENCES .....	91
INITIAL DISTRIBUTION LIST .....	92

## I. INTRODUCTION

A visual aid for students new to orbital mechanics is required to comprehend fully the dynamics of orbital motion. This program is an interactive time step simulation program that calculates and plots either unperturbed or perturbed elliptical orbits. The program interacts with the student in developing the initial orbit. Also the program enables the student with the ability to change the velocity of the satellite at a specific location in the orbit. This feature will permit the student to investigate the effects of commanded velocity changes as in perigee kicks, apogee kicks and inclination changes. The user can also modify the initial position and velocity of the satellite at the completion of any orbit.

The student is given an opportunity to investigate the effects of perturbing forces on the satellites orbit by choosing to have the program calculate the orbit with or without perturbing forces. The variation of parameters method, as seen in [Ref. 1; pp. 396-407], is used in calculating the perturbing orbit. The perturbing forces taken into consideration are the following:

1. the oblateness of the earth
2. drag for low earth orbits
3. gravitational force of the moon
4. gravitational force of the sun

In order to review fully the operation of the program (included in appendix A) and to uncover any problems or limitations that plagued the programming, the program has been divided up as follows:

1. program design
2. unperturbed orbit
3. perturbed orbit
4. velocity changes
5. graphical plots

The programming approach and equations used in each of the above sections will be examined in there respective chapters. A review of the coordinate systems used and their

transformations between them are included in appendix B. Since all the equations used in the calculation of the orbital elements are from reference 1, they will not be reviewed in each chapter but will be included in appendix C for a quick reference. Equations from other sources will be referenced in their respective chapters.

Examples of perturbed and unperturbed orbital plots for a variety of initial orbital parameters are included in appendix D. Included are plots of low earth orbits, transfer orbits and geosynchronous orbits.

## II. PROGRAM DESIGN

In designing this program an attempt was made to make it not only as user friendly as possible, but also to make the program as simple as possible to understand. To achieve these goals, the program would have to be written in a logical manner, in a computer language that is easy to follow, the program would have to run on terminals readily available to students (at the Naval Postgraduate School (NPS)), and the program would have to be easily used by students with a minimum amount of computer or orbital mechanics knowledge.

FORTRAN was chosen as the programming language since it is a widely used scientific language and it allows for very structured programming. By programming in a structured format, the program can be expanded in the future with a minimum amount of time required to understand the programming code. FORTRAN also allows for double precision numbers to be used in the calculation of the orbit. This is critical when round off error in single precision could be greater than the actual change that one is trying to model. The equations in the descriptions of the program might not exactly match the equations in the listings because of special programming techniques which must be included in most computer programs to handle such problems as "division by zero".

The display integrated software system and plotting language (DISSPLA) package available on the mainframe computer at NPS was used to enable a variety of graphical displays with a minimum amount of programming. DISSPLA has a set of subroutines that the programmer calls to display data contained in arrays. This requirement forces the program to load arrays with the satellites position in order for it to be plotted. The TEC618 computer terminal and associative plotter was used for ease of gaining hard copy plots of the orbits and the diversity of locations that are available here at NPS. In order to run a program in DISSPLA the user must first define storage space of 1500K and designate temporary disk space, and then call DISSPLA with the program name. This is accomplished with the following commands:

1. DEFINE STORAGE 1500K

2. ICMS
3. TDISK 4 DIS
4. DISPLAY ORBIT

To make the program user friendly, the user is prompted for inputs via the keyboard. The entry is usually a number. A yes or no response can be entered by typing "Y" or a "N". In most cases the program does a check to see if the input is appropriate. In order to make it as easy as possible for the student to get the desired orbit displayed, the program requires only the initial position and velocity of the satellite. The initial position and velocity of the satellite is supplied by the user in one of two ways. The user can input the position and velocity of the satellite, using the perifocal coordinate system (IJK), or the user can let the program place the satellite on the "I" axis of the IJK system at the radius of perigee (RP) distance supplied by the user. This latter choice gives the initial location of the satellite, but to get the velocity the program will prompt the user for one of the following:

1. the actual velocity in the IJK system.
2. the eccentricity (e) of the orbit. In which case the velocity is calculated from the following equations:

$$a = \frac{RP}{1-e} = \text{semi-major axis}$$

$$ENR = -\frac{\mu}{2a} = \text{energy mass}$$

Where  $\mu = MG$

$M = \text{mass of earth}$

$G = \text{Universal gravitational constant}$

$$r = \sqrt{2(ENR + \frac{\mu}{RP})}$$

3. the radius of apogee (RA). The velocity is calculated by first calculating the eccentricity (e) from the following:

$$e = \frac{RA - RP}{RA + RP}$$

With the eccentricity the same equations used above are used to calculate the velocity.

In order to give the velocity a direction the inclination (i) of the orbit is required from the user. The following equations are used to calculate the velocity vector:

$$v_I = 0.0$$

$r_p = 1.28911$

$r_a = 1.51811$

The program will check to ensure that the orbital eccentricity is less than 1.0, if it is not then the program will reject the inputs. After the initial input are accepted, the program will do calculations for the six orbital elements required to describe the size, shape and orientation of the orbit, and to pinpoint the position of the satellite along the orbit at a particular time. This classical set of six orbital elements are as follows:

1. a, semi-major axis.
2. e, eccentricity.
3. i, inclination.
4.  $\Omega$ , longitude of the ascending node.
5.  $\omega$ , argument of perigee passage.
6. T, time of perigee passage.

The program actually calculates more orbital elements than the six classical elements required to plot the orbit, this is done in an effort to make the program as robust as possible. This will add in the ability to expand the program in the future.

If the satellite is not initially at the perigee point then the satellite will first be stepped around to the perigee point. The program then enters a loop that calculates the orbit from the perigee point through one complete orbit around the earth and back to the perigee point. The orbit is calculated in steps of 2 times pi divided by an integer, i.e., 2 times pi divided by 50. This step size was used to ensure a smooth orbit for display purposes and also to get within adequate distance to the perigee point or other location for a velocity change. After the loop is completed, the program will offer the user a choice of the following plots to check the orbit:

1. perifocal
2. groundtrack

The program then goes into a loop offering the user the following choices until the user decides to end the program:

1. plot another view of the same orbit.

If the user wishes to plot another view of the same orbit then the user may use this choice to reenter the display portion of the program.

2. plot the next orbit (perturbed or unperturbed).

To plot the next orbit the satellite is stepped around the complete orbit either with or without perturbing forces effecting the satellite.

3. change the initial conditions.

The program goes to the beginning of the program and allows the user to change the initial position and velocity of the satellite.

4. change the velocity at a specific location

Step the satellite around to a specific true anomaly and make a velocity change at that location.

5. clear the previous orbits from the plot.

Clear the memory of all the previous orbits and only retain the current location and velocity as the initial position and velocity.

Before each new orbit, the orbital elements are recalculated.

There are several common assumptions and constants used throughout the program i.e. all bodies are considered to be spherically symmetric (this allows these bodies to be treated as though their masses are concentrated at their centers (point masses)). other assumptions will be covered in their respective chapters.

### III. UNPERTURBED ORBIT

The subroutines that calculate the unperturbed orbit are the most widely used subroutines in the entire program. These subroutines are called to step the satellite around to the perigee point from the user supplied initial position and velocity, to calculate the next unperturbed orbit, and for any velocity change. No matter which of these sources supply the initial position and velocity the program calculates the unperturbed orbit in the same manner. The only difference is where in the orbit the satellite is initially when these subroutines are called. Before the unperturbed subroutines are called, the orbital elements are calculated.

The unperturbed subroutines are called by a single subroutine "UNPRET" which has the following basic algorithm:

1. Increment time by the time step size (DT). The time step was chosen as the period divided by fifty to give a smooth plot, but more importantly to ensure that the satellite is within an acceptable distance from a specific location for a velocity change. The angular error caused by the step size can be as much as PI/50 from the desired point for a circular orbit and will increase for more eccentric orbits. This error becomes a factor when the user is making velocity changes, and therefore it will be covered in that chapter in further detail.
2. Calculate the new elements. The calculation of the new elements is the heart of this algorithm. The size, shape and orientation of the orbit remains unchanged. What is required is the position of the satellite along the orbit as a function of time. The problem becomes a matter to solve "the Kepler problem"-predicting the future position and velocity of an orbiting object as a function of some known initial position and velocity and the time of flight [Ref. I: p. 181]. An algorithm using these principles will follow:
  - a. A time step (DT) is added to the time of flight(TF), time of flight is the elapsed time since the satellite passed the perigee point.  
$$TF = TF + DT$$
  - b. The new mean anomaly (MA) is calculated from the new time of flight, and the mean motion (MM).  
$$MA = MM \times TF$$
  - c. With the new mean anomaly the new eccentric anomaly (EA) is calculated. Because the solution to the Kepler problem ( $M.A = E.A - e \times \sin(E.A)$ ) is transcendental, an iterative solution based on the Newton method of root finding is used. The root in question is a solution to the equation ( $M.A - E.A + e \times \sin(E.A) = 0$ ). This algorithm takes the form of [Ref. I: p. 222]:
    - 1)  $M.A = E.A - e \times \sin(E.A)$

2.

$$E.I_{\omega_1} = E.I_n - \frac{(M.I - M.A_n)}{(1 - e \times \cos(E.I_n))}$$

Where this equation is applied initially to  $E.I_n = M.I$  and then reapplied until the difference between  $M.A$  and  $M.I$ , becomes small enough to be ignored.

d. The new true anomaly ( $v_r$ ) is calculated from:

$$v_r = \frac{\cos^{-1}(e - \cos(E.I))}{e \cos(E.I) - 1}$$

3. Calculate the new position and velocity. The position and velocity are calculated in the perifocal coordinate system (PQW). The PQW system uses the orbit as its fundamental plane and therefore requires only two coordinate to specify the satellite's position and velocity. The  $z_n$  coordinate is by definition always equal to zero. The position of the satellite is calculated as:

$$x_w = r \cos v$$

$$y_w = r \sin v$$

$$z_n = 0$$

The velocity of the satellite is calculated as:

$$v_r = \sqrt{\frac{\mu}{r}} ( - \sin v_r )$$

$$v_t = \sqrt{\frac{\mu}{r}} ( e + \cos v_r )$$

$$v_z = 0$$

4. Store position and elements in arrays for plotting. In order for the program to plot the orbit the radius, true anomaly, inclination, and argument of perigee must be stored in arrays. The use of these arrays to plot the orbit will be explained in chapter 6.
5. The process is repeated until the satellite is at the perigee point and the true anomaly is two pi.

The procedure used to calculate the unperturbed orbit leave very little to be modified by a programmer. The only choices that had to be made concerned step size, how to tell the UNFRET subroutine that the perigee point had been reached, and a value of acceptable error for newtons method. For the unperturbed orbit, the step size just had to be small enough to produce a smooth plot of the orbit. Two indicators for perigee were used, one was that the true anomaly was greater than 6.21 radians (two pi equals 6.28 radians) and the time from the previous perigee point will be greater then the period. The two indicators were logically 'and' together to ensure the perigee point was reached.

The disparity between two pi and 6.21 radians is due to the error produced by the satellite not beginning the orbit at exactly the perigee point and the step size used go around the orbit. The acceptable size of error for newtons method was set at  $1.0 \times 10^{-10}$ , because for an unperturbed orbit this would be the major contributor to any error in the orbit and the magnitude of this error would be acceptable. However; in a perturbed orbit there are other factors contributing to determining the acceptable error, and these will be discussed in the next chapter.

#### IV. PERTURBED ORBIT

The perturbed orbit uses the same basic routines as the unperturbed orbit in stepping the satellite around the earth with one major difference, the perturbing forces produce a time rate of change of the orbital elements that must be applied at each time step. The variation of parameters method is used to determine this influence of the perturbing forces on the orbital elements. The analysis is simplified by using the orbital coordinate system 'RSW', as explained in appendix B. The basic algorithm is as follows [Ref. 1: p. 407]:

1. At  $t = t_0$ , calculate six orbital elements.
2. Compute the perturbing forces and transform it at  $t = t_0$  to the "RSW" SYSTEM.
3. Compute the time rate-of-change of the elements.
4. Calculate the change of elements for one time step, and add the changes to the old values at each step to get the new elements.
5. From the new values of the orbital elements, calculate a position and velocity.
6. Go to the step 2 and repeat until the final time is reached.

The steps in the algorithm will be explained in the following sections:

##### A. ORBITAL ELEMENTS

The standard orbital elements  $a$ ,  $e$ ,  $i$ ,  $\Omega$ ,  $\omega$  and  $T$  (or  $M$ ) will be used, where

$a$  = semi-major axis

$e$  = eccentricity

$i$  = inclination

$\Omega$  = longitude of ascending node

$\omega$  = argument of perigee

$T$  = time of perigee passage

( $M_e$  = mean anomaly at epoch =  $M - n(t - t_0)$ ). The elements are calculated only at the beginning of the orbit from the initial position and velocity vectors. The elements are then changed continuously throughout the orbit by adding the changes due to the perturbing forces. For the perturbed orbit, the satellite will always begin at the perigee point. This is done so one complete orbit is from perigee point to perigee point.

## B. COMPUTE PERTURBING FORCES

The variation of parameters method requires that the perturbing forces be calculated at each step in the orbit. In order to do this a model of each perturbing force must be developed. The following perturbing forces were used in calculating the total perturbing force effecting the satellite:

1. oblateness of the earth
2. atmospheric drag
3. gravitational attraction of the sun
4. gravitational attraction of the moon

The magnitudes of these forces have an enormous range of values and are dependent on the distance the satellite is from the perturbing body. Figure 1 on page 12 shows a graphical representation of the magnitude of the perturbing forces in a log-log plot of perturbing forces per unit mass [Ref. 2: p. IV-61]. The model of each of these forces follows:

### I. NON-SPHERICAL EARTH

The earth is not perfectly spherical, but bulges around the equator. The polar and equatorial diameters are 12713.0 Km and 12756.3 Km, respectively. The oblateness results in a perturbing force per unit mass with these components in the "RSW" coordinate system [Ref. 3: p. 51]:

$$F_r = \frac{(-3\mu J_2 r_e^2)}{2r^2} (1 - 3 \sin^2(i) \sin^2(u_3))$$

$$F_t = \frac{(-3\mu J_2 r_e^2)}{r^2} (\sin^2(i) \sin(u_3) \cos(u_3))$$

$$F_w = \frac{(-3\mu J_2 r_e^2)}{r^2} (\sin(i) \cos(i) \sin(u_3))$$

The variable and constants of these equations are defined below:

#### I. Variables:

- a.  $u_3$  = the argument of latitude and is equal to the true anomaly  $v_0$  plus the argument of perigee  $\omega$ .

$$u_3 = v_0 + \omega$$

- b.  $r$  = the radius from the center of the earth to the satellite.

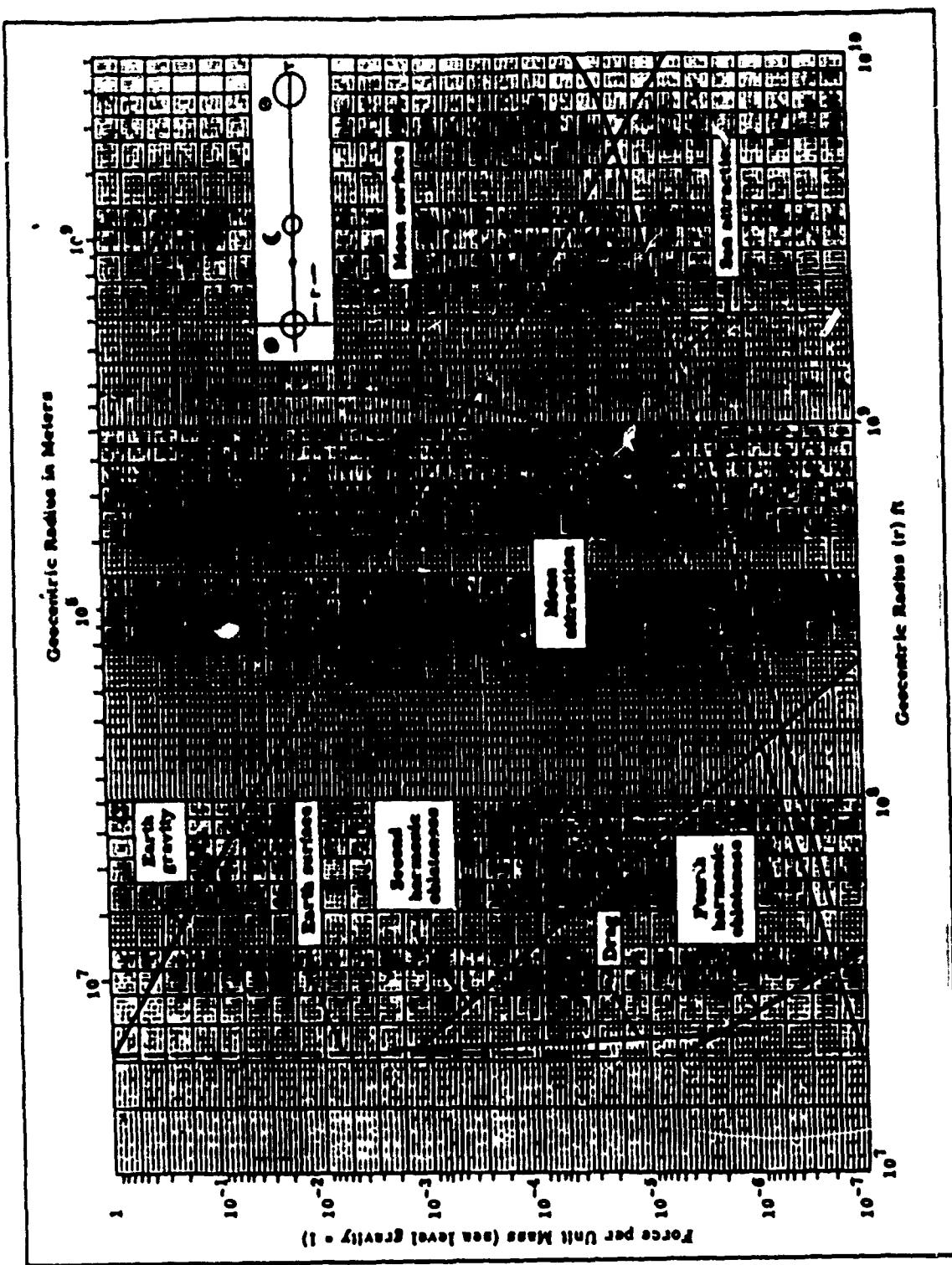


Figure 1. Comparison of perturbation magnitudes.

$$r = \tilde{r}$$

## 2. Constants:

- a.  $\mu$  = the gravitational parameter of the earth.

$$\mu = 398601.2 \frac{(km^3)}{s^2}$$

- b.  $J_2$  = the second harmonic of oblateness coefficient, determined by experimental observations.

$$J_2 = 1.0823E - 3$$

- c.  $r_e$  = the mean radius of the earth,

$$r_e = 6.3782E3 km$$

## 2. ATMOSPHERIC DRAG

The formulation of atmospheric drag equations are plagued with uncertainties of atmospheric fluctuations, frontal areas of orbiting object (if not constant), the drag coefficient, and other parameters. A fairly simple formulation will be given here. Drag, by definition, will be opposite to the velocity of the vehicle relative to the atmosphere. Thus, the perturbing force is

$$\tilde{F} = -\left(\frac{1}{2m}\right) \cdot CD \cdot IR \cdot DEN \cdot v \cdot \tilde{v}$$

The velocity vector is in the 'IJK' system so the resulting force is also in the 'IJK' system. Therefore a transformation to the 'RSW' system is required.

The variables and constants of this equation are defined below:

### I. Variables:

- a.  $v$  = speed of vehicle.

- b.  $CD$  = the dimensionless drag coefficient. The drag coefficient  $CD$  has a value between 1 and 2. It takes a value near 1 when the mean free path of the atmospheric molecules is small compared with the satellite size, and takes a value close to 2 when the mean free path is large compared with the size of the satellite. The drag coefficient will be modeled with  $CD = 2$  when the satellites altitude is greater than 550km and equal to 1 otherwise. [Ref. 4: p. 295]

- c.  $DEN$  = atmospheric density at the vehicle's altitude. The density is spherically symmetric, and will be modeled using exponential steps using the parameters in Table 1 on page 14 and the following formula [Ref. 1: pp. 423-424]:  
$$\delta(z) = \delta_0 e^{-z/z_0}$$
.

Table 1. ATMOSPHERIC PARAMETERS AND VALUES

$k_{\text{B}} \cdot \beta$	$\lambda$	$\gamma$	$\delta \cdot r$
0.150	1.223E.02	4.74E.02	0.0
			150
150.55	1.79840E.01	4.3614E.02	550
550	1.015484E.07	2.21698E.07	1500
			4100
			1.0E+12

2. Constants set to typical values:

- a.  $m$  = mass of the satellite, set equal to 100kg.
- b.  $AR$  = the sectional area of the vehicle perpendicular to the direction of motion.

3. PERTURBATION DUE TO HEAVENLY BODY

The satellite experiences perturbation forces due to the gravitational effects of the sun and the moon. The perturbation force from a perturbing body is the difference between the gravitational force due to the perturbing body at the satellite and the gravitational force the satellite would experience if it were at the center of the earth. From Figure 2 on page 15, the perturbing force per unit mass of the satellite is

$$f_p = \mu_p \frac{\vec{r}_p \vec{i}_p - \vec{r}_s \vec{i}_s}{|\vec{r}_p \vec{i}_p - \vec{r}_s \vec{i}_s|^3} - \frac{\mu_s \vec{r}_s}{|\vec{r}|^3}$$

The variable and constants are defined below:

1. Variables:

- a.  $r_p$  = distance from the earth center for the perturbing body
- b.  $\vec{i}_p$  = unit vector from the earth to the perturbing body
- c.  $r$  = distance from earth center to the satellite
- d.  $\vec{i}_s$  = unit vector from the earth to the satellite

2. Constants:

- a.  $\mu_p$  = gravitational constant of the perturbing body =  $M_p G$

The subscript p is to be replaced by s if the perturbing body is the sun, and by m if the perturbing body is the moon. We will assume that  $r \ll r_p$ , then the equation above becomes

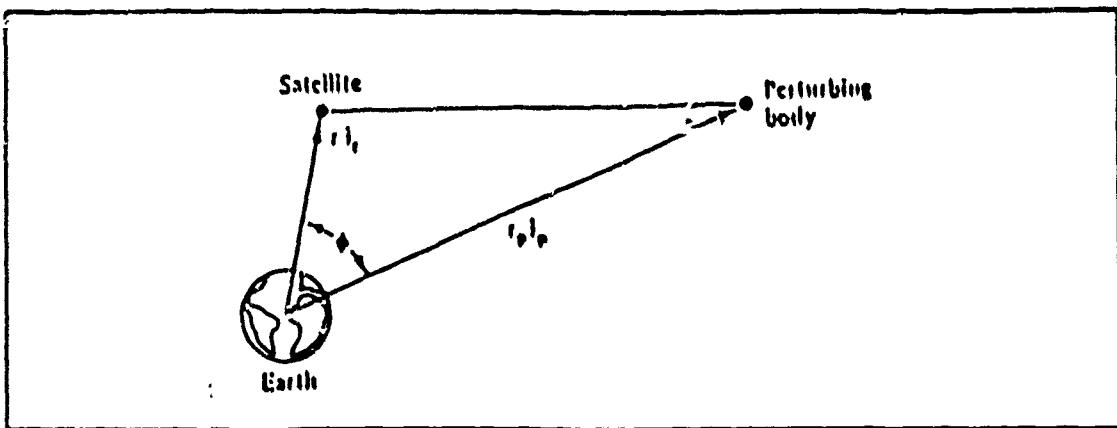


Figure 2. Perturbation forces.

$$\vec{F}_p = \left( \frac{\mu_p}{r_p^3} \right) \left( \frac{r}{r_p} \right) \left( 3(\vec{i}, \vec{i}_p) \vec{i}_p - \vec{i}_r \right)$$

The unit vectors  $\vec{i}_r$  and  $\vec{i}_p$  can be written in terms of the 'IJK' system as:

$$\vec{i}_r = (\cos(\Omega) \cos(u_0) - \sin(\Omega) \cos(i) \sin(u_0)) \vec{I} + (\cos(u_0) \sin(\Omega) + \cos(\omega) \cos(i) \sin(u_0)) \vec{J} + (\sin(i) \sin(u_0)) \vec{K}$$

$$\vec{i}_p = (\cos(\Omega_p) \cos(u_{0p}) - \sin(\Omega_p) \cos(i_p) \sin(u_{0p})) \vec{I} + (\cos(u_{0p}) \sin(\Omega_p) + \cos(\omega_p) \cos(i_p) \sin(u_{0p})) \vec{J} + (\sin(i_p) \sin(u_{0p})) \vec{K}$$

where  $\Omega$ ,  $i$ , and  $u_0$  are the orbital elements of the satellites and  $\Omega_p$ ,  $i_p$ , and  $u_{0p}$  are the orbital elements of the perturbing body. The formulas above use the 'IJK' system, and as such the resultant forces must be transformed to the 'RSW' system. Models of the sun and moon orbits are required to calculate  $\vec{r}$ , and  $\vec{i}_r$ . The models used in the program for the sun and moon's orbits follows: [Ref. 3: pp. 73-74]

#### a. SUN'S POSITION

In order to model the sun's orbit, a number of simplifications had to be made in the actual parameters of the sun's orbit. First the sun will be assumed to be in a circular orbit. This means that the radius ( $r$ ) to the sun will be constant, and the eccentricity ( $e$ ) will equal 0.0 instead of its true value of 0.017. The other assumption will

be to place the sun on the 'I' axis of the 'IJK' system at the beginning of the program and have it progress through its orbit as the program runs. These changes will not effect the perturbing force in any noticeable magnitude.

The following variables and constants where used in the program to model the sun's orbit after applying the simplifications: [Ref. 3: pp. 75-78]

### 1. Constants:

a. Gravitational Constant:  $G = 6.67E - 11 \frac{\text{Nm}^2}{\text{kg}^2}$

b. Sun's Mass:  $m_s = 1.99E30 \text{kg}$

c. Sun's Gravitational parameter:

$$\mu_s = 1.32733E20 \frac{\text{Nm}^2}{\text{kg}}$$

d. Sun's eccentricity:  $e_s = 0.0$

e. Radius of orbit, assume sun is in circular orbit:  $r_s = 1.49E11 \text{m}$

f. Sun's inclination:  $si = 23.45 \text{ deg.} = 4.09279709d-01 \text{ radians}$

g. Longitude of ascending node:  $\Omega_s = 0.0$

h. Argument of perigee:  $\omega_s = 0.0$

### 2. Variables:

a. The true anomaly of the sun's position as a function of the time the satellite has been in orbit:

$$v_o(TT) = \frac{2\pi}{356 \times 24 \times 3600} TT$$

Where TT = true time, the time the satellite has been in orbit (sec)

b. Sun's Position vector:  $\vec{r}_s = r \cos v_o \vec{P} + r \sin v_o \vec{Q}$

c. Unit vector from the earth to the sun:  $\hat{i}_s = \frac{\vec{r}_s}{|\vec{r}_s|}$

### b. MOON'S POSITION

In modeling the orbit of the moon, similar assumptions where used as with the sun. The moons orbit will be assumed to be circular, actually the eccentricity is equal to 0.055. By placing the moon initially on the 'I' axis of the 'IJK' system along with the sun, the gravitational forces of the two bodies will combine to a maximum. However; since the moons orbital period is only 27.3 days, the moon will not stay in this alignment and the magnitude of the combined forces will vary with time. The inclination of the moons orbit is not constant, but drifts between 18.3 and 28.6 degrees in ten years.

Also the longitude of the ascending node ( $\Omega$ ) oscillates between 13 and -13 degrees. To simplify this the inclination will be chosen as a constant 23.5 degrees and the longitude of the ascending node at 0.0 degrees. For the time period involved in calculating the perturbed orbit, these assumptions will not make any significant difference.

The following variables and constants were used in the program to model the moon's orbit, after applying the simplifications:

### 1. Constants:

- a. Gravitational Constant:  $G = 6.67E-11 \frac{(Nm^2)}{kg^2}$
- b. Moon's Mass:  $m_m = 7.35E22 kg$
- c. Moon's Gravitational Parameter:  $\mu_m = GM_m = 4.90E12 \frac{(Nm^2)}{kg}$
- d. Moon's eccentricity:  $e_m = 0.0$
- e. Radius of orbit, assume moon is in circular orbit:  $r_m = 3.844E8 km$
- f. Moon's inclination:  $i = 23.5\text{deg.} = 4.10152374E-1 \text{ radians}$
- g. Moon's longitude of ascending node:  $\Omega_m = 0.0$
- h. Moon's argument of perigee:  $\omega_m = 0.0$
- i. Moon's period:  $T = 27.3 \text{ days [period]}$

### 2. Variables:

- a. The true anomaly of the moon's position as a function of the time the satellite has been in orbit:  $v_m(7T) = \frac{2\pi}{27.3 \times 24 \times 3600} 7T$
- b. Moon's position Vector:  $\vec{r}_m = r \cos v_m \vec{P} + r \sin v_m \vec{Q}$
- c. Unit vector from earth to moon:  $\hat{r}_m = \frac{\vec{r}_m}{|\vec{r}_m|}$

The models of the sun and moon's orbit calculates the position vector in the 'PQW' system and therefore the position vector must be transformed to the 'IJK' system.

## C. RATE-OF-CHANGE OF ORBITAL ELEMENTS

The derivations and equations of the rates-of-change of the orbital elements are contained in reference 1 pages 398 to 406. Therefore; only a summary of the actual analytic expressions for the rate-of-change of the parameters in terms of the perturbations will follow:

### 1. Rate-of-change of the semi-major axis:

$$\frac{d\dot{r}}{dt} = \left[ \frac{2e \sin v_0}{n' \sqrt{1-e^2}} \right] F_r - \left[ \frac{2\dot{r} \sqrt{1-e^2}}{n'r} \right] F_t$$

Where  $n'$  is the mean motion of the satellite's orbit.

$$n' = \sqrt{\frac{1}{a^3}}$$

## 2. Rate-of-change of the eccentricity:

$$\frac{de}{dt} = \left[ \frac{\sqrt{1-e^2} \sin v_0}{n'a} \right] F_r + \left[ \frac{\sqrt{1-e^2}}{n'a^2 e} \right] \left[ \frac{a^2(1-e^2)}{r} - r \right] F_t$$

## 3. Rate-of-change of the inclination:

$$\frac{di}{dt} = \left[ \frac{r \cos u_0}{n'a^2 \sqrt{1-e^2}} \right] F_w$$

## 4. Rate-of-change of the longitude of the ascending node:

$$\frac{d\Omega}{dt} = \left[ \frac{r \sin u_0}{n'a^2 \sqrt{1-e^2} \sin i} \right] F_n$$

## 5. Rate-of-change of the argument of perigee:

$$\frac{d\omega}{dt} = (d \frac{v}{dt})_r - (d \frac{v}{dt})_t + (d \frac{v}{dt})_w$$

Where,

$$(d \frac{v}{dt})_r = \left[ \frac{-\sqrt{1-e^2} \cos v_0}{n'a e} \right] F_r$$

$$(d \frac{v}{dt})_t = \left[ \frac{r}{e h} \right] \left[ \sin v_0 \left( 1 + \frac{1}{1+e \cos v_0} \right) \right] F_t$$

$$(d \frac{v}{dt})_w = \left[ \frac{-r \cot i \sin u_0}{n'a^2 \sqrt{1-e^2}} \right] F_w$$

## 6. Rate-of-change of the eccentric anomaly:

$$\frac{dE.A}{dt} = \frac{1}{\sin(E.A)} \frac{\left[ (\sin v_0 + \frac{de}{dt})(1+e \cos v_0) - (\cos v_0 + e)(\frac{de}{dt} \cos v_0 + e \sin v_0) \right]}{\left[ 1+e \cos v_0 \right]^2}$$

## 7. Rate-of-change of the mean anomaly:

$$\frac{dM.A}{dt} = \frac{dE.A}{dt} - \frac{de}{dt} \sin(E.A) - e \times \cos \frac{(E.A)dE.A}{dt} - \frac{dn'}{dt} (t - t_0)$$

This equation reduces to the following for circular and elliptic orbits ( $m_s = e^2 - 1$ ):

$$\frac{dM/F}{dt} = \frac{-1}{n/a} \left[ \frac{2r}{a} - \frac{(1-e^2)}{e} \cos v_r \right] F_r - \left[ \frac{1-e^2}{n^2 a^2} \right] \left[ 1 - \frac{r}{a(1-e^2)} \right] \sin v_r F_t + t \frac{dv'}{dt}$$

Where the Rate-of-change of the mean motion:

$$\frac{dv'}{dt} = \left[ \frac{-3\mu}{2n'^2 a^2} \right] \frac{da}{dt}$$

[ref. 1 p. 396-407]

#### D. NEW ORBITAL ELEMENTS

The change of each element is calculated by multiplying the rate-of-change of the element by the time step (DT). The change in the orbital elements are then added to the current values of the elements to give the new orbital elements. With the new elements calculated, the satellite is stepped forward and the new position and velocity are calculated in the same manner as the unperturbed orbit (chapter 3). Also as with the unperturbed orbit, the process is repeated until the satellite is at the perigee point, indicated by the time of flight (TF) equal to the period of the perturbed orbit.

## V. VELOCITY CHANGES

The ability of the student to change the velocity of the satellite at any position in the orbit is a vital element in this program. With velocity changes the student can investigate the effects of varying the satellites velocity as in transfer orbits and inclination changes. In order to simplify the program the unperturbed orbit is used throughout this routine. The velocity change algorithm used in the program follows:

### 1. Rotate to velocity change location.

The user is given the choice of changing the velocity of the satellite at the perigee, apogee or at any true anomaly. If the user chooses perigee or apogee as the change locations, the true anomaly is set equal to zero or pi radians respectively. With the location of the velocity change, the satellite is first stepped around to the desired true anomaly. The stepping is identical with the unperturbed orbit with the exception that the stepping terminates when the true anomaly is greater or equal to the desired true anomaly. With a step size of one fiftieth of the period, the satellite is actually stepped around to a location near the desired location. This variance can be reduced by decreasing the step size but this would increase the computation time. This error will be a major factor in precise calculations of transfer orbits, or any other orbital maneuver where precise velocity changes are required. However, this program is not a tool to calculate precise orbital maneuvers, but rather a learning tool for the student to get a feel for the results of velocity changes in a satellite's orbit.

### 2. Change the velocity.

With the satellite at the desired location, the program calculates and displays for the user the satellite's current velocity, escape velocity and circular velocity (the velocity required to circularize the orbit). The program will not allow velocities greater than or equal to the escape velocity. The user is given the option to enter a new velocity in the 'IJK' system or to change the magnitude of the velocity in the orbital plane. If the user chooses to change the velocity in the orbital plane, the program will prompt the user for the magnitude of the velocity change, and multiply this change by a unit vector in the direction of the satellites' velocity. This velocity change vector is then added to the satellites velocity vector, to calculate the new velocity vector.

### 3. Calculate new elements.

The orbital elements are calculated with the new velocity vector and the satellite's position vector.

### 4. Complete the orbit.

The program will complete the orbit to the new perigee point using the satellite's position, new velocity and new elements. There are a number of problems that arise if the satellite is just stepped around to the perigee point. For example, with velocity changes in the orbital plane the apogee and perigee directions can physically swap. This is a problem when plotting with the perifocal coordinate system because the  $X_1$  axis points toward perigee. To avoid problems like this the arrays used in plotting the orbit must be cleared and the satellite's current position

and velocity be treated as initial conditions. However, to compare the old and new orbits there is a desire to retain as much of the previous orbit as possible. The velocity changes were divided into the following four cases to handle these problems:

- a. Change velocity in the orbital plane at the perigee point with the new velocity greater than the circular velocity. The perigee point will remain the same so the satellite is stepped around using the unperturbed subroutines.
- b. Change velocity in the orbital plane at the perigee point with the new velocity less than or equal to the circular velocity. The perigee and apogee directions will switch so the plotting arrays are first cleared and stored with the current location data. Because the satellite is now at the apogee point the satellite is stepped around to the perigee point storing the second half of the orbit. The entire next orbit is calculated and stored to get a complete orbit.
- c. Change velocity in the orbital plane at the apogee point with the new velocity less than the circular velocity. The perigee and apogee directions will remain the same, so the satellite is stepped around to the perigee point completing the orbit.
- d. This last case catches all the following velocity changes; velocity change in the orbital plane at the apogee point with the new velocity greater than or equal to the circular velocity, velocity changes at any other true anomaly in the orbital plane, and any velocity change out of the orbital plane. The plotting arrays are cleared and stored with the current location data. No matter where in the orbit the satellite is, the satellite is first stepped around to the perigee point, and to ensure a complete orbit is plotted the entire next orbit is also calculated and stored.

## VI. GRAPHICAL PLOTS

The program provides two types of graphical displays of the orbit, a display in the perifocal coordinate system and a display of the satellite's ground track. Each display type is useful in observing different aspects of the orbit. The perifocal display will allow the user to see how certain orbital parameters change with different initial positions and velocities, and also how the parameters change with velocity changes at varying positions in the orbit. The ground track will enable the user to gain an appreciation for the physical location of the satellite above the earth, and see how the orbital parameter affects the path of the satellite. The ground track will also display the precession of a sequence of orbits. Both displays plot the position steps to give the user an understanding of how the satellite speeds up at perigee and slows down around apogee.

The DISSPLA package on the mainframe computer was used to enable the plotting of the orbits. The versatility of plotting subroutines of DISSPLA makes the actual programming of the orbit a simple matter of initializing DISSPLA for the type of monitor being used, setting up the plotting area, initializing the axis and axis scale, and then plotting the desired curve from points contained in arrays. This is a simplified explanation of DISSPLA, but for further details on DISSPLA programming refer to the DISSPLA user's manual [Ref. 5]. DISSPLA also supplies subroutines to draw a variety of projections of the world and fill the projections with coast lines, latitude lines and longitude lines. There are a couple of DISSPLA requirements that did require special handling in the program. The requirement that the data be supplied in arrays forced the program to load arrays with the required position and parameters and to keep a counter for the number in the arrays. The array format requires the size of the array be specified in the beginning of the program. The array size needs to be large enough to hold a number of orbits, but not so large as to waste storage space. The program will continue to add orbital data to the arrays until the user chooses to delete the previous orbits. If a new initial position and velocity is entered or if the arrays will overflow with the next orbit the arrays will automatically delete all previous orbits. DISSPLA also requires that all data be in single precision format. The program calculates all orbits in double precision in order to limit the effect of round-off error, but by using the single precision data for plotting will not affect the accuracy of the plot in any way.

The subroutines used to display the orbits will be covered in the following three sections:

#### A. PERIFOCAL PLOT

The plotting of the orbit in the perifocal coordinate system is the easier of the two types of plots. Since the perifocal coordinate system has the orbital plane as the fundamental plane, the only requirements to describe the orbit in the perifocal coordinate system are arrays with the true anomaly and the radius to the satellite. To give the user a sense of the size of the plot, the axis length varies with the eccentricity and semi-major axis length. Also a plot of the earth is plotted to the same scale, with the pole or center of the plot on the origin of the axis. The latitude of the earth at the center of the plot will vary with the inclination of the orbit. This plot will allow the user to see a relative view of the satellite's coverage in the minus 'Z' axis direction of the perifocal coordinate system.

#### B. GROUND TRACK

The ground track plot is a very complex subroutine compared with the perifocal plot. Because the ground track is not a continuous curve a procedure to handle the satellite ending at one end of the plot and wrapping around to the other end was developed. The wrap around problem is avoided in most orbits by plotting the orbit in segments with the following two rules. Each segment begins at the beginning of a new plot or at the edge of the plot area, and ending when the satellite would wrap around to the other side of the plot. At the beginning of a segment if the position of the satellite is within five degrees of the edge of the plot, that position and any other positions within that five degree boundary will not be plotted. The segment will end when the satellite is within ten degrees of the edge of the plot. The above restrictions imposed on the segments of the plot will not substantially affect the interpretation or usefulness of the plot. The ground track is plotted on top of a cylindrical equidistant projection of the world, with the world coast lines and a longitude-latitude grid for reference.

#### C. DATA

Information concerning the orbit is displayed on the lower half of the plot. The information is designed to supply the user with enough of the basic orbital elements and other parameters affecting the orbit to be able to evaluate what basic type of orbit the satellite is in, and the effects of velocity changes and perturbing forces have on the orbit. The following data are plotted: inclination(i), semi-major axis (a), eccentricity (e), period

(peri), apogee and perigee velocity and radius, average time rate-of-change of orbital elements, and the average magnitude of perturbing forces per unit mass.

## VII. CONCLUSIONS AND RECOMMENDATIONS

The program supplies the student with an interactive tool to study the orbital motion of satellites around the earth. The student can investigate a variety of orbits by varying the orbital parameters, command velocity changes, and observe the effects of perturbing forces.

The student is provided with two options for entering the initial position and velocity of the satellite. The program could be expanded to provide the student with the additional options of entering either orbital parameters or a ground observation data and have the program calculate the initial position and velocity from this data. Also the student is limited to orbits with eccentricities less than one (elliptic orbits). The program could be also be expanded to include more eccentric orbit for Lunar, interplanetary, and missile trajectories. The perturbing orbit is calculated for orbits around the earth with relatively small perturbing forces in relation to the earth's gravitational force. This fact will cause the program to produce false results if the student tries to calculate lunar trajectories. Special routines would have to be employed when the perturbing force (the moon's gravitational attraction) is comparable to the earth's gravitational attraction. This will not become a factor for studying current satellite orbits out to the geosynchronous radius of 42241.1km.

The velocity change subroutines move the satellite to a location close to the desired location before a velocity change is imposed. By reducing the step size in the velocity change subroutine, this error could be reduced. Precise orbital transfer maneuvers can be modeled by reducing this error caused by the positioning of the satellite prior to changing the velocity. The program will currently provide the student with useful plots for gaining experience with various transfer orbits by varying the magnitude and location of the velocity changes.

The output of the calculations of the orbit are arrays loaded with the satellite's position and select orbital parameters. The DISSPLA subroutines that plot the points are not unique. The program would become portable to personal computers with these graphics subroutines written in FORTRAN and included in the program.

A final recommendation is that the display of the ground track could be modified to show ground coverage, number of satellites in a constellation, and other elements necessary for planning a real-world artificial satellite application.

## APPENDIX A. ORBIT PROGRAM

### PROGRAM ORBIT

THIS PROGRAM IS AN INTERACTIVE TIME STEP SIMULATION OF SATELLITES AROUND THE EARTH. PERTURBED AND UNPERTURBED ORBITS ARE CALCULATED AND PLOTTED. VELOCITY CHANGES ARE ALSO PERMITTED AT SPECIFIED TRUE ANOMALIES.

A LIST OF VARIABLES USED BY THE MAIN PROGRAM FOLLOWS.

ORB00010	
ORB00020	
ORB00030	
ORB00040	
ORB00050	
ORB00060	
ORB00070	
ORB00080	
ORB00090	
ORB00100	
ORB00110	
ORB00120	
ORB00130	
ORB00140	
ORB00150	
ORB00160	
ORB00170	
ORB00180	
ORB00190	
ORB00200	
ORB00210	
ORB00220	
ORB00230	
ORB00240	
ORB00250	
ORB00260	
ORB00270	
ORB00280	
ORB00290	
ORB00300	
ORB00310	
ORB00320	
ORB00330	
ORB00340	
ORB00350	
ORB00360	
ORB00370	
ORB00380	
ORB00390	
ORB00400	
ORB00410	
ORB00420	
ORB00430	
ORB00440	
ORB00450	
ORB00460	
ORB00470	
ORB00480	
ORB00490	
ORB00500	
ORB00510	

★	TDE	= TOTAL CHANGE IN ECCENTRICITY	ORB00520
★	TDH	= TOTAL CHANGE IN ANGULAR MOMENTUM	ORB00530
★	TDI	= TOTAL CHANGE IN INCLINATION	ORB00540
★	TDMA	= TOTAL CHANGE IN MEAN ANOMALY	ORB00550
★	TDMM	= TOTAL CHANGE IN MEAN MOTION	ORB00560
★	TDLAN	= TOTAL CHANGE IN LONGITUDE OF ASCENDING NODE	ORB00570
★	TF	= TIME OF FLIGHT	ORB00580
★	TFDRA	= TOTAL FORCE OF DRAG	ORB00590
★	TFEA	= TOTAL FORCE OF EARTH'S OBLATENESS	ORB00600
★	TFMO	= TOTAL FORCE FROM MOON	ORB00610
★	TFSU	= TOTAL FORCE FROM SUN	ORB00620
★	TL	= TRUE Longitude AT EPOCH	ORB00630
★	TT	= TRUE TIME SINCE SATELLITE HAS BEEN IN ORBIT	ORB00640
★	V	= SATELLITE VELOCITY	ORB00650
★	VI	= I VECTOR OF SATELLITE VELOCITY	ORB00660
★	VJ	= J VECTOR OF SATELLITE VELOCITY	ORB00670
★	VK	= K VECTOR OF SATELLITE VELOCITY	ORB00680
★			ORB00690
★		A LIST OF THE ARRAYS USED FOLLOWS:	ORB00700
★			ORB00710
★	AINRAY	= INCLINATION	ORB00720
★	APRAY	= ARGUMENT OF PERIGEE	ORB00730
★	RARAY	= RADIUS	ORB00740
★	RIRAY	= I VECTOR OF RADIUS	ORB00750
★	RJRAY	= J VECTOR OF RADIUS	ORB00760
★	RKRAY	= K VECTOR OF RADIUS	ORB00770
★	TARAY	= TRUE ANOMALY	ORB00780
★	TIMRAY	= TIME	ORB00790
★			ORB00800
★		A LIST OF SUBROUTINES CALLED BY THE MAIN PROGRAM WILL FOLLOW:	ORB00810
★			ORB00820
★	CALCEL	= CALCULATES THE ORBITAL ELEMENTS	ORB00830
★	CHGVEL	= ALLOW THE USER TO CHANGE THE VELOCITY OF THE SATELLITE	ORB00840
★	INPUTS	= PROMPTS USER FOR INITIAL POSITION AND VELOCITY	ORB00850
★	INTSUM	= INITIALIZES THE SUMS IN THE ARRAYS	ORB00860
★	NEWELT	= CALCULATE NEW ORBITAL ELEMENTS FROM TIME STEP	ORB00870
★	NEWPOS	= CALCULATE NEW POSITION VECTOR	ORB00880
★	NEWVEL	= CALCULATE NEW VELOCITY VECTOR	ORB00890
★	OPTION	= GIVE THE USER THE OPTIONS Permitted IN THE PROGRAM	ORB00900
★	PLOTS	= PLOTS THE ORBITS	ORB00910
★	PRETUR	= CALCULATES THE PERTURBED ORBIT	ORB00920
★	STORE	= STORE THE POSITION DATA IN ARRAYS	ORB00930
★	UNPRET	= CALCULATE THE UNPERTURBED ORBIT	ORB00940
★			ORB00950
★	BEGIN MAIN PROGRAM		ORB00960
★			ORB00970
+	DOUBLE PRECISION PI,MU,RI,RJ,RK,R,VI,VJ,VK,V,HI,HJ,HK,H,		ORB00980
+	NI,NJ,NK,N,P,EI,EJ,EK,E,A,I,LAN,AP,TA,AL,LP,TL,PER,EA,		ORB00990
+	MM,MA,T,DT,TF, FR,FS,FW,TT,CHTA,RA,VA,TEMPTA,RE		ORB01000
+			ORB01010
+	DIMENSION TARAY(500),RARAY(500),RIRAY(500),RJRAY(500),RKRAY(500),		ORB01020
+	AINRAY(500),APRAY(500),TIMRAY(500)		ORB01030
	CHARACTER*1,LOOP,YORN,ORLOOP		ORB01040
	PI = 3.141592653589794		ORB01050
			ORB01060
			ORB01070

MU = 3.986012D+05	ORB01080
RE = 6.378145D+03	ORB01090
* USER INTRO TO PROGRAM	ORB01100
CALL INTRO	ORB01110
* ENTERED MAIN PROGRAM LOOP	ORB01120
LOOP = 'Y'	ORB01130
10 IF (LOOP .EQ. 'Y') THEN	ORB01140
* INITIALIZE STEP COUNTER AND TRUE TIME	ORB01150
20 NUM = 1	ORB01160
TT = 0.0	ORB01170
* PROMPT USER FOR INITIAL POSITION AND VELOCITY	ORB01180
CALL INPUTS(RI,RJ,RK,R,VI,VJ,VK,V,MU,LOOP,PI)	ORB01190
* EXIT PROGRAM	ORB01200
IF (LOOP .EQ. 'N') THEN	ORB01210
GOTO 10	ORB01220
ENDIF	ORB01230
* CALCULATE AND STORE ORBITAL ELEMENTS	ORB01240
CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,	ORB01250
LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)	ORB01260
+ CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,	ORB01270
NUM,I,AP,AIRRAY,APRAY,TT,TIMRAY)	ORB01280
+ PRINT DATE FOR USER TO REVIEW	ORB01290
PRINT*, 'VI = ', VI, ' KM/S'	ORB01300
PRINT*, 'VJ = ', VJ, ' KM/S'	ORB01310
PRINT*, 'VK = ', VK, ' KM/S'	ORB01320
PRINT*, 'V = ', V, ' KM/S'	ORB01330
PRINT*, 'RI = ', RI, ' KM'	ORB01340
PRINT*, 'RJ = ', RJ, ' KM'	ORB01350
PRINT*, 'RK = ', RK, ' KM'	ORB01360
PRINT*, 'R = ', R, ' KM'	ORB01370
PRINT*, 'ECCENTRICITY = ', E	ORB01380
DEGI = SNGL((180.0/PI)*I)	ORB01390
PRINT*, 'INCLINATION = ', DEGI, ' DEGREES'	ORB01400
PERHRS = SNGL(PER/3600.0)	ORB01410
PRINT*, 'PERIOD = ', PERHRS, ' HOURS'	ORB01420
PRINT*, 'ARE THESE VALUES CORRECT? ENTER "Y" OR "N" : '	ORB01430
READ*, YORN	ORB01440
CALL EXCMS('CLRSCRN')	ORB01450
IF (.NOT. YORN .EQ. 'Y') THEN	ORB01460
GOTO 20	ORB01470
ENDIF	ORB01480
* CALCULATE TIME STEP AND SET TIMER TO ONE TIME STEP	ORB01490
DT = PER/50	ORB01500
T = DT	ORB01510
* STEP SATELLITE TO PERIGEE POINT AND RECORD	ORB01520
50 IF ((TA.GT.0.063).AND.(TA.LT.6.21)) THEN	ORB01530
TT = TT + DT	ORB01540
	ORB01550
	ORB01560
	ORB01570
	ORB01580
	ORB01590
	ORB01600
	ORB01610
	ORB01620
	ORB01630

```

CALL NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER)          ORE01640
CALL NPOS(RI,RJ,RK,R,LAN,AP,I,TA,A,E)           ORB01650
CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU)        ORB01660
NUM = NUM + 1                                     ORB01670
CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,    ORB01680
          NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)          ORB01690
T = T + DT                                       ORB01700
GOTO 50                                         ORB01710
ENDIF                                           ORB01720
*      CALCULATE ELEMENTS FROM PERIGEE POINT      ORB01730
CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,    ORB01740
          LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)    ORB01750
DT = PER/50                                      ORB01760
T = DT                                           ORB01770
*      STORE FIRST Unperturbed ORBIT              ORB01780
CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,    ORB01790
          MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,T,NUM,RIRAY,RJRAY,    ORB01800
          RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY,TT)          ORB01810
*      INITIALIZE SUMS FOR FORCE AND ORBITAL ELEMENT CHANGES TO ZERO   ORB01820
CALL INTSUM(TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDMY,TDMA,TDLAN,    ORB01830
          TDH,TDAP)                                     ORB01840
*      PLOT FIRST UNPERTURBED ORBIT                ORB01850
70      CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM,PI,I,LP,A,E,TF,    ORB01860
          +                                             AIRRAY,APRAY,TIMRAY,TFEA,TFSU,TFMO,TFDRA,PER,TDI,TDA,    ORB01870
          +                                             TDE,TDMY,TDMA,TDLAN,TDH,TDAP,MM,MA,LAN,H,AP,R,V)    ORB01880
          +                                             ORB01890
*      BEGIN NEW ORBIT OPTIONS                    ORB01900
*      IOPT1 = 1. Unperturbed ORBIT             ORB01910
*      = 2. Perturbed ORBIT                   ORB01920
*      = 3. QUIT                           ORB01930
*      IOPT2 = 1. PLOT NEXT ORBIT             ORB01940
*      = 2. CHANGE INITIAL VALUES          ORB01950
*      = 3. CHANGE VELOCITY AT A SPECIFIC TRUE Anomaly    ORB01960
*      = 4. PLOT ANOTHER VIEW OF SAME ORBIT    ORB01970
*      ALSO ASKED IF WANT TO CLEAR ALL PREVIOUS ORBITS    ORB01980
*      CALCULATE ELEMENTS AT PERIGEE            ORB01990
80      CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,    ORB02000
          +                                             LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)    ORB02010
*      CHECK FOR POSSIBLE ARRAY OVERFLOW        ORB02020
IF (NUM .GT. 425) THEN                         ORB02030
  PRINT*, 'ARRAYS ARE FULL'                  ORB02040
  PRINT*, 'PREVIOUS ORBITS WILL BE ERASED!'    ORB02050
  NUM = 1                                     ORB02060
  CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,    ORB02070
          +                                             NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)          ORB02080
  ENDIF                                         ORB02090
*      PROMPT USER FOR DESIRED OPTION          ORB02100

```

```

CALL OPTION(IOPT1,ICPT2,NUM,RIRAY,RJRAY,RARAY,
TARAY,AIRRAY,APRAY,TIMRAY) ORB02200
+
Initialize SUMS FOR FORCE AND ORBITAL ELEMENT CHANGES TO ZERO ORB02210
CALL INTSUM(TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDM,TDMA,TDLAN, ORB02220
TDH,TDAP) ORB02230
+
SET TIME COUNTER TO ONE TIME STEP ORB02240
T = DT ORB02250
+
OPTION: PLOT THE NEXT ORBIT ORB02260
IF (IOPT2 .EQ. 1) THEN ORB02270
+
CALCULATE AND PLOT UNPERTURBED ORBIT ORB02280
IF(IOPT1 .EQ. 1) THEN ORB02290
+
CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ, ORB02300
RK,R,VI,VJ,VK,V,MU,PI,H,A, ORB02310
E,N,TA,P,MM,MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY, ORB02320
RARAY,TARAY,AIRRAY,APRAY,TIMRAY,TT) ORB02330
+
CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB02340
PI,I,LP,A,E,TF,AIRRAY,APRAY,TIMRAY, ORB02350
TFEA,TFSU,TFMO,TFDRA,PER, ORB02360
TDI,TDA,TDE,TDM,TDMA,TDLAN,TDH,TDAP, ORB02370
MM,MA,LAN,H,AP,R,V) ORB02380
+
CALCULATE AND PLOT PERTURBED ORBIT ORB02390
ELSEIF(IOPT1 .EQ. 2) THEN ORB02400
+
CALL PRETUR(DT,PER,AL,LAN,AP,I, ORB02410
RI,RJ,RK,R,VI,VJ,VK,V,FR,FS,Fw, ORB02420
MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,T,NUM, ORB02430
RIRAY,RJRAY,RKRAY,RARAY,TARAY,AIRRAY,APRAY, ORB02440
TIMRAY,TT,TFEA,TFSU,TFMO,TFDRA, ORB02450
TDI,TDA,TDE,TDM,TDMA,TDLAN,TDH,TDAP) ORB02460
+
CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB02470
PI,I,LP,A,E,TF,AIRRAY,APRAY,TIMRAY, ORB02480
TFEA,TFSU,TFMO,TFDRA,PFR, ORB02490
TDI,TDA,TDE,TDM,TDMA,TDLAN,TDH,TDAP, ORB02500
MM,MA,LAN,H,AP,R,V) ORB02510
ENDIF ORB02520
+
GOTO THE BEGINNING OF THE PROGRAM TO CHANGE THE INITIAL VALUES ORB02530
ELSEIF (IOPT2 .EQ. 2) THEN ORB02540
GOTO 20 ORB02550
+
CHANGE VELOCITY AT A SPECIFIC TRUE ANOMALY AND ORB02560
PLOT THE NEW ORBIT ORB02570
+
ELSEIF (IOPT2 .EQ. 3) THEN ORB02580
+
CALL CHGVEL(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R, ORB02590
VI,VI,VK,V,MU,PI, ORB02600
H,A,E,N,TA,P,MM,MA,EA,TF,T,NUM,RIRAY, ORB02610
RJRAY,RKRAY,RARAY,TARAY,AIRRAY,APRAY, ORB02620
TIMRAY,TT,EI,EJ,EK,LP,HI,HJ,IOPT1, ORB02630
TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDM, ORB02640
TDMA,TDLAN,TDH,TDAP) ORB02650
+
CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB02660
PI,I,LP,A,E,TF,AIRRAY,APRAY,TIMRAY, ORB02670
TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDM, ORB02680
TDMA,TDLAN,TDH,TDAP) ORB02690
+
CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB02700
PI,I,LP,A,E,TF,AIRRAY,APRAY,TIMRAY, ORB02710
TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDM, ORB02720
TDMA,TDLAN,TDH,TDAP) ORB02730
+
CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB02740
PI,I,LP,A,E,TF,AIRRAY,APRAY,TIMRAY, ORB02750
TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDM, TDMA,TDLAN,TDH,TDAP)

```

```

+      TFEA,TFSU,TFMO,TFDRA,PER,          CRB02760
+      TDI,TDA,TDE,TDM4,TDMA,TDLAN,TDN,TDAP, ORB02770
+      MM,MA,LAN,H,AP,R,V)                CRB02780
+      ORB02790
*      PLOT ANOTHER VIEW OF THE SAME ORBIT      ORB02800
ELSEIF (IOP2 .EQ. 4) THEN          ORB02810
    CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB02820
+      PI,I,LP,A,E,TF,AIRAY,APRAY,TIMRAY,        ORB02830
+      TFEA,TFSU,TFMO,TFDRA,PER,                  ORB02840
+      TDI,TDA,TDE,TDM4,TDMA,TDLAN,TDN,TDAP,      ORB02850
+      MM,MA,LAN,H,AP)                         ORB02860
+      ORB02870
*      STOP THE PROGRAM                      ORB02880
ELSEIF (IOP2 .EQ. 5) THEN          ORB02890
    GOTO 90                                ORB02900
ELSE                                ORB02910
    PRINT*, 'INVALID ENTRY!'             ORB02920
    GOTO 80                                ORB02930
ENDIF                               ORB02940
ORB02950
*      CHECK IF SATELLITE Impacted THE EARTH AND GO TO THE BEGINNING      ORB02960
IF (R .LE. 6450.0) THEN          ORB02970
    PRINT*, 'SATELLITE WILL IMPACT THE EARTH!!!'          ORB02980
    PRINT*, 'PROGRAM WILL RESET TO THE BEGINNING!'        ORB02990
    GOTO 20                                ORB03000
ENDIF                               ORB03010
ORB03020
*      GOTO THE TOP OF THE OPTION LOOP                      ORB03030
GOTO 60                                ORB03040
ORB03050
*      GIVE THE USER A CHANCE TO RECOVER THE PROGRAM          ORB03060
90   PRINT*, 'THIS IS YOUR LAST CHANCE!'          ORB03070
    PRINT*, 'DO YOU WANT TO CONTINUE?'          ORB03080
    PRINT*, 'AND GOTO THE Beginning OF THE PROGRAM?'        ORB03090
    PRINT*, 'ENTER "Y" OR "N" : '              ORB03100
    READ*,LOOP                            ORB03110
    PRINT*,LOOP                            ORB03120
    GOTO 10                                ORB03130
ENDIF                               ORB03140
ORB03150
*      DISSPLA SUBROUTINE TO TELL GRAPHICS TERMINAL PLOTTING      ORB03160
**      SESSION IS DONE
    CALL DONEPL                            ORB03170
    STOP                                 ORB03180
    END                                  ORB03190
ORB03200
ORB03210
*****SUBROUTINE INTRO*****
*      THIS SUBROUTINE WILL GIVE THE USER A Brief INTRODUCTION OF THE      ORB03220
**      USES OF THE PROGRAM                                ORB03230
ORB03240
*      PRINT*, 'THIS PROGRAM IS A GRAPHICS DISPLAY OF Satellite ORBITS.'      ORB03250
PRINT*, 'YOU WILL BE ASKED TO INPUT THE INITIAL VELOCITY AND'          ORB03260
PRINT*, 'POSITION VECTORS OF THE Satellite. THE PROGRAM WILL '          ORB03270
PRINT*, 'THEN CALCULATE THE ORBITAL PARAMETERS AND THE '          ORB03280
ORB03290
ORB03300
ORB03310

```

```

PRINT*, 'Unperturbed ORBIT. THE USER WILL THEN HAVE THE' ORB03320
PRINT*, 'CHOICE OF DISPLAYS:' ORB03330
PRINT*, '    -PERIFOCAL (SHOWS RELATIVE SIZE OF ORBIT)' ORB03340
PRINT*, '    -Equatorial (SHOWS ORBIT INCLINED, USER INPUT' ORB03350
PRINT*, '        LONGITUDE TO VIEW AT)' ORB03360
PRINT*, '    -GROUND TRACK' ORB03370
PRINT*, 'THE USER IS THEN ASKED TO CHOOSE ONE OF THE FOLLOWING:' ORB03380
PRINT*, '    -Unperturbed ORBITS' ORB03400
PRINT*, '    -Perturbed ORBITS' ORB03410
PRINT*, '    -VELOCITY CHANGES' ORB03420
PRINT*, 'THE USER'S CHOICE WILL BE USED IN DEVELOPING THE' ORB03430
PRINT*, 'GRAPHICAL OUTPUT.' ORB03440
PRINT*, 'THE USER IS THEN GIVEN THE FOLLOWING CHOICES:' ORB03450
PRINT*, '    -CLEAR ALL THE PREVIOUS ORBITS' ORB03460
PRINT*, '    -CHANGE THE INITIAL PARAMETERS' ORB03480
PRINT*, '    -CHANGE VELOCITY AT A SPECIFIC TRUE Anomaly' ORB03490
PRINT*, '    -PLOT ANOTHER VIEW OF THE SAME ORBIT' ORB03500
RETURN ORB03510
END ORB03520
***** ORB03530
SUBROUTINE OPTION(IOPT1,IOPT2,NUM,RIRAY,RJRAY,RKRAY,RARAY, ORB03540
+      TARAY,AIRAY,APRAY,TIMRAY) ORB03550
* THIS SUBROUTINE GIVES THE USER A CHOICE OF OPERATIONS THAT CAN BE ORB03560
* PERFORMED ON THE PROGRAM AND RETURNS THE USERS CHOICE WITH ORB03570
* VARIABLES IOPT1 AND IOPT2 ORB03580
DIMENSION RIRAY(500),RJRAY(500),RKRAY(500),RARAY(500),TARAY(500), ORB03610
+      AIRAY(500),APRAY(500),TIMRAY(500) ORB03620
CHARACTER*1,YORN ORB03630
IOPT1 = 0 ORB03640
ORB03650
ORB03660
* PROMPT USER FOR OPTION ORB03670
103 PRINT*, 'WHICH OF THE FOLLOWING OPTIONS WOULD YOU LIKE:' ORB03680
PRINT*, '    1. -CALCULATE THE NEXT ORBIT USING THE SAME' ORB03690
PRINT*, '        PARAMETERS' ORB03700
PRINT*, '    2. -CHANGE THE INITIAL PARAMETERS OF THE ORBIT' ORB03710
PRINT*, '    3. -CHANGE THE VELOCITY AT A POINT IN THE ORBIT' ORB03720
PRINT*, '        (THE UNPERTURBED ORBIT WILL BE USED)' ORB03730
PRINT*, '    4. -PLOT ANOTHER VIEW OF THE ORBIT(S)' ORB03740
PRINT*, '    5. -QUIT' ORB03750
PRINT*, 'ENTER 1, 2, 3, 4, OR 5:' ORB03760
READ*,IOPT2 ORB03770
PRINT*,IOPT2 ORB03780
CALL EXCMN('CLRSCRN') ORB03790
IF ( IOPT2 .GT. 5 ) THEN ORB03800
GOTO 103 ORB03810
ENDIF ORB03820
ORB03830
* Prompt USER FOR TYPE OF ORBIT DESIRED ORB03840
105 IF ( IOPT2 .EQ. 1 ) THEN ORB03850
PRINT*, 'WHICH TYPE OF ORBIT WOULD YOU LIKE TO SEE,' ORB03860
PRINT*, '    1. -Unperturbed ORBITS' ORB03870

```

```

PRINT*, ' 2.-Perturbed ORBITS'          ORB03880
PRINT*, ' ENTER 1 OR 2:'                ORB03890
READ*, IOPT1                           ORB03900
PRINT*, IOPT1                           ORB03910
CALL EXCMS('CLRSCRN')                 ORB03920
IF ((IOPT1 .NE. 1) .AND. (IOPT1 .NE. 2)) THEN
    PRINT*, 'INVALID ENTRY!'           ORB03930
    GOTO 105                           ORB03940
ENDIF                                ORB03950
ENDIF                                ORB03960
* PROMPT USER TO CLEAR PREVIOUS ORBITS   ORB03970
107 IF ((IOPT2 .EQ. 1) .OR. (IOPT2 .EQ. 3)) THEN   ORB03980
    PRINT*, 'DO YOU WANT TO CLEAR THE PREVIOUS ORBITS?' ORB04000
    PRINT*, 'ENTER "Y" OR "N" :'        ORB04010
    READ*, YORN                         ORB04020
    PRINT*, YORN                         ORB04030
    CALL EXCMS('Clrscren')              ORB04040
    IF (YORN .EQ. 'Y') THEN            ORB04050
        KIRAY(1) = RIRAY(NUM)          ORB04060
        RJRAY(1) = RJRAY(NUM)          ORB04070
        RKRAY(1) = RKRAY(NUM)          ORB04080
        RARAY(1) = RARAY(NUM)          ORB04090
        TARAY(1) = TARAY(NUM)          ORB04100
        AINRAY(1) = AINRAY(NUM)         ORB04110
        APRAY(1) = APRAY(NUM)          ORB04120
        TIMRAY(1) = TIMRAY(NUM)         ORB04130
        NUM = 1                          ORB04140
    ELSEIF (YORN .NE. 'N') THEN       ORB04150
        PRINT*, 'INVALID ENTRY!'       ORB04160
        PRINT*, 'ALL INPUTS MUST BE CAPITOL LETTERS' ORB04170
        GOTO 107                         ORB04180
    ENDIF                                ORB04190
ENDIF                                ORB04200
* CHECK FOR INVALID OPTION             ORB04210
IF ((IOPT2 .NE. 1) .AND. (IOPT2 .NE. 2) .AND. (IOPT2 .NE. 3) .AND.
+     (IOPT2 .NE. 4) .AND. (IOPT2 .NE. 5)) THEN   ORB04220
    PRINT*, 'INVALID ENTRY!'           ORB04230
    GOTO 103                           ORB04240
ENDIF                                ORB04250
RETURN                               ORB04260
END                                  ORB04270
***** * COORDINATE TRANSFORMATIONS ***** ORB04280
* SUBROUTINE PQWIJK(LAN,AP,INC,P,Q,W,I,J,K)      ORB04290
* THIS SUBROUTINE TRANSFORMS PQW COORDINATES TO IJK COORDINATES ORB04300
* DOUBLE PRECISION INC,P,Q,W,I,J,K,R11,R12,R13,R21,R22,R23, ORB04310
+     R31,R32,R33,LAN,AP                  ORB04320
    R11 = DCOS(LAN)*DCOS(AP) - DSIN(LAN)*DSIN(AP)*DCOS(INC) ORB04330
    R12 = -DCOS(LAN)*DSIN(AP) - DSIN(LAN)*DCOS(AP)*DCOS(INC) ORB04340
    R13 = DSIN(LAN)*DSIN(INC)           ORB04350

```

```

R21 = DSIN(LAN)*DCOS(AP) + DCOS(LAN)*DSIN(AP)*DCOS(INC) ORB04440
R22 = -DSIN(LAN)*DSIN(AP) + DCOS(LAN)*DCOS(AP)*DCOS(INC) ORB04450
R23 = -DCOS(LAN)*DSIN(INC) ORB04460
R31 = DSIN(AP)*DSIN(INC) ORB04470
R32 = DCOS(AP)*DSIN(INC) ORB04480
R33 = DCOS(INC) ORB04490
I = R11*I + R12*J + R13*K ORB04500
J = R21*I + R22*J + R23*K ORB04510
K = R31*I + R32*J + R33*K ORB04520
RETURN ORB04530
END ORB04540
ORB04550
ORB04560
ORB04570
ORB04580
ORB04590
ORB04600
ORB04610
ORB04620
ORB04630
ORB04640
ORB04650
ORB04660
ORB04670
ORB04680
ORB04690
ORB04700
ORB04710
ORB04720
ORB04730
ORB04740
ORB04750
ORB04760
ORB04770
ORB04780
ORB04790
ORB04800
ORB04810
ORB04820
ORB04830
ORB04840
ORB04850
ORB04860
ORB04870
ORB04880
ORB04890
ORB04900
ORB04910
ORB04920
ORB04930
ORB04940
ORB04950
ORB04960
ORB04970
ORB04980

```

\*\*\*\*\*  
\* SUBROUTINE IJKPOW(LAN,AP,INC,I,J,K,P,Q,W)  
\* THIS SUBROUTINE TRANSFORMS IJK COORDINATES TO PQW COORDINATES

```

DOUBLE PRECISION INC,I,J,K,P,Q,W,R11,R12,R13,R21,R22,R23,
+ R31,R32,R33,LAN,AP
R11 = DCOS(LAN)*DCOS(AP) - DSIN(LAN)*DSIN(AP)*DCOS(INC) ORB04630
R21 = -DCOS(LAN)*DSIN(AP) - DSIN(LAN)*DCOS(AP)*DCOS(INC) ORB04640
R31 = DSIN(LAN)*DSIN(INC) ORB04650
R12 = DSIN(LAN)*DCOS(AP) + DCOS(LAN)*DSIN(AP)*DCOS(INC) ORB04660
R22 = -DSIN(LAN)*DSIN(AP) + DCOS(LAN)*DCOS(AP)*DCOS(INC) ORB04670
R32 = -DCOS(LAN)*DSIN(INC) ORB04680
R13 = DSIN(AP)*DSIN(INC) ORB04690
R23 = DCOS(AP)*DSIN(INC) ORB04700
R33 = DCOS(INC) ORB04710
P = R11*I + R12*J + R13*K ORB04720
Q = R21*I + R22*J + R23*K ORB04730
W = R31*I + R32*J + R33*K ORB04740
RETURN ORB04750
END ORB04760
ORB04770
ORB04780
ORB04790
ORB04800
ORB04810
ORB04820
ORB04830
ORB04840
ORB04850
ORB04860
ORB04870
ORB04880
ORB04890
ORB04900
ORB04910
ORB04920
ORB04930
ORB04940
ORB04950
ORB04960
ORB04970
ORB04980

```

\*\*\*\*\*  
\* SUBROUTINE IJKRSW(LAN,AL,INC,I,J,K,R,S,W)  
\* THIS SUBROUTINE CHANGES FROM IJK COORDINATES TO RSW COORDINATES

```

DOUBLE PRECISION INC,I,J,K,R,S,W,R11,R12,R13,R21,R22,R23,
+ R31,R32,R33,LAN,AL
R11 = DCOS(LAN)*DCOS(AL) - DSIN(LAN)*DCOS(INC)*DSIN(AL) ORB04850
R12 = DSIN(LAN)*DCOS(AL) + DSIN(AL)*DCOS(LAN)*DCOS(INC) ORB04860
R13 = DSIN(INC)*DSIN(AL) ORB04870
R21 = -DCOS(LAN)*DSIN(AL)-DSIN(LAN)*DCOS(INC)*DCOS(AL) ORB04880
R22 = -DSIN(LAN)*DSIN(AL) + DCOS(LAN)*DCOS(INC)*DCOS(AL) ORB04890
R23 = DSIN(INC)*DCOS(AL) ORB04900
R31 = DSIN(LAN)*DSIN(INC) ORB04910
R32 = -DCOS(LAN)*DSIN(INC) ORB04920
R33 = DCOS(INC) ORB04930
R = R11*I + R12*J + R13*K ORB04940
S = R21*I + R22*J + R23*K ORB04950
W = R31*I + R32*J + R33*K ORB04960
RETURN ORB04970
END ORB04980

```

```

***** SUBROUTINE RSWIJK(LAN,AL,INC,R,S,W,I,J,K) ***** ORB04990
* THIS SUBROUTINE CHANGES FROM RSW COORDINATES TO IJK COORDINATES ORB05000
* DOUBLE PRECISION INC,R,S,W,I,J,K,R11,R12,R13,R21,R22,R23, ORB05010
+ R31,R32,R33,LAN,AL ORB05020
  R11 = DCOS(LAN)*DCOS(AL) - DSIN(LAN)*DCOS(INC)*DSIN(AL) ORB05030
  R21 = DSIN(LAN)*DCOS(AL) + DSIN(AL)*DCOS(LAN)*DCOS(INC) ORB05040
  R31 = DSIN(INC)*DSIN(AL) ORB05050
  R12 = -DCOS(LAN)*DSIN(AL)-DSIN(LAN)*DCOS(INC)*DCOS(AL) ORB05060
  R22 = -DSIN(LAN)*DSIN(AL) + DCOS(LAN)*DCOS(INC)*DCOS(AL) ORB05070
  R32 = DSIN(INC)*DCOS(AL) ORB05080
  R13 = DSIN(LAN)*DSIN(INC) ORB05090
  R23 = -DCOS(LAN)*DSIN(INC) ORB05100
  R33 = DCOS(INC) ORB05110
  I = R11*R + R12*S + R13*W ORB05120
  J = R21*R + R22*S + R23*W ORB05130
  K = R31*R + R32*S + R33*W ORB05140
  RETURN ORB05150
  END ORB05160
***** SUBROUTINE PQWRSW(TA,P,Q,W,R,S,WN) ***** ORB05170
* THIS SUBROUTINE CHANGES FROM PQW COORDINATES TO RSW COORDINATES ORB05180
* DOUBLE PRECISION P,Q,W,R,S,WN,R11,R12,R13,R21,R22,R23, ORB05190
+ R31,R32,R33,TA ORB05200
  R11 = DCOS(TA) ORB05210
  R12 = DSIN(TA) ORB05220
  R13 = 0.0 ORB05230
  R21 = -DSIN(TA) ORB05240
  R22 = DCOS(TA) ORB05250
  R23 = 0.0 ORB05260
  R31 = 0.0 ORB05270
  R32 = 0.0 ORB05280
  R33 = 1.0 ORB05290
  R = R11*P + R12*Q + R13*W ORB05300
  S = R21*P + R22*Q + R23*W ORB05310
  WN = R31*P + R32*Q +R33*W ORB05320
  RETURN ORB05330
  END ORB05340
***** SUBROUTINE RSWPQW(TA,R,S,W,P,Q,WN) ***** ORB05350
* THIS SUBROUTINE CHANGES FROM RSW COORDINATES TO PQW COORDINATES ORB05360
* DOUBLE PRECISION R,S,W,P,Q,WN,R11,R12,R13,R21,R22,R23, ORB05370
+ R31,R32,R33,TA ORB05380
  R11 = DCOS(TA) ORB05390
  R21 = DSIN(TA) ORB05400
  R31 = 0.0 ORB05410
  R12 = -DSIN(TA) ORB05420

```

```

R22 = DCOS(TA) ORB05550
R32 = 0.0 ORB05560
R13 = 0.0 ORB05570
R23 = 0.0 ORB05580
R33 = 1.0 ORB05590
P = R11*R + R12*S + R13*W ORB05600
Q = R21*R + R22*S + R23*W ORB05610
WN = R31*R + R32*S + R33*W ORB05620
RETURN ORB05630
END ORB05640
ORB05650
*****
* STORE ELEMENTS IN ARRAYS ORB05660
*****
SUBROUTINE STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM, ORB05700
+ I,AP,AIRRAY,APRAY,TT,TIMRAY) ORB05710
* THIS SUBROUTINE STORES THE POSITION AND ELEMENTS IN ARRAYS IN ORB05720
* SINGLE PRECISION FORM, FOR PLOTTING ORB05730
DOUBLE PRECISION RI,RJ,RK,R,TA,I,AP,TT ORB05740
DIMENSION RIRAY(500),RJRAY(500),RKRAY(500),RARAY(500),TARAY(500), ORB05770
+ AIRRAY(500),APRAY(500),TIMRAY(500) ORB05780
ORB05790
RIRAY(NUM) = SNGL(RI) ORB05800
RJRAY(NUM) = SNGL(RJ) ORB05810
RKRAY(NUM) = SNGL(RK) ORB05820
RARAY(NUM) = SNGL(R) ORB05830
TARAY(NUM) = SNGL(TA) ORB05840
AIRRAY(NUM) = SNGL(I) ORB05850
APRAY(NUM) = SNGL(AP) ORB05860
TIMRAY(NUM) = SNGL(TT) ORB05870
RETURN ORB05880
END ORB05890
ORB05900
*****
* INITIAL POSITION, VELOCITY ORB05910
*****
SUBROUTINE INPUTS(RI,RJ,RK,R,VI,VJ,VK,V,MU,QUIT PI) ORB05950
* THIS SUBROUTINE GIVES THE USER A CHOICE TO ENTER THE ORB05960
* INITIAL POSITION AND VELOCITY VECTOR OR TO LET THE PROGRAM ORB05970
* CALCULATE THE INITIAL POSITION AND VELOCITY BASED ON PROMPTED ORB05980
* INPUTS ORB05990
ORB06000
* SUBROUTINES CALLED FROM THIS SUBROUTINE: ORB06010
* INELTS = Prompts user for orbital elements ORB06020
* IPPOS = PROMPTS user for initial position (IJK) ORB06030
* IVEL = PROMPTS user for initial Velocity (IJK) ORB06040
ORB06050
DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,MU,PI ORB06060
CHARACTER*1,QUIT ORB06070
ORB06080
* PROMPT USER FOR METHOD TO ENTER INPUTS ORB06090
195 PRINT*: 'IN WHICH MANNER WOULD YOU LIKE TO INPUT THE INITIAL' ORB06100

```

```

PRINT*, 'POSITION AND VELOCITY OF THE SATELLITE?' ORB06110
PRINT*, ' 1: BY Inputting THE INITIAL POSITION AND VELOCITY' ORB06120
PRINT*, '    VECTORS IN THE PERIFOCAL COORDINATE SYSTEM (IJK)' ORB06130
PRINT*, ' 2: BY LETTING THE SATELLITE BE PLACED ON THE "I"' ORB06140
PRINT*, '    AXIS OF THE (IJK) SYSTEM AT A DESIRED RADIUS OF' ORB06150
PRINT*, '    PERIGEE(RP) AND INPUTTING EITHER A DESIRED RADIUS' ORB06160
PRINT*, '    OF APOGEE(RA), A DESIRED ECCENTRICITY(E), OR THE' ORB06170
PRINT*, '    DESIRED VELOCITY AT THAT RADIUS, AND A DESIRED' ORB06180
PRINT*, '    INCLINATION(I).' ORB06190
PRINT*, ' 3: QUIT' ORB06200
PRINT*, 'ENTER 1, 2 OR 3:' ORB06210
READ*, ICHC ORB06220
PRINT*, ICHC ORB06230
CALL EXCMS('CLRSCRN') ORB06240
ORB06250

* USER INPUTS POSITION AND VELOCITY VECTORS ORB06260
IF (ICHC .EQ. 1) THEN ORB06270
  CALL IFOS(RI,RJ,RK,R) ORB06280
  CALL IVEL(VI,VJ,VK,V,R,MU) ORB06290
ORB06300

* USER INPUTS ORBITAL ELEMENTS TO GET POSITION AND VELOCITY ORB06310
ELSEIF (ICHC .EQ. 2) THEN ORB06320
  CALL INELTS(RI,RJ,RK,R,VI,VJ,VK,V,MU,PI) ORB06330
ORB06340

* STOP PROGRAM ORB06350
ELSEIF (ICHC .EQ. 3) THEN ORB06360
  QUIT = 'N' ORB06370
ELSE ORB06380
  PRINT*, 'INVALID ENTRY! TRY AGAIN!' ORB06390
  GOTO 195 ORB06400
ENDIF ORB06410
RETURN ORB06420
END ORB06430
ORB06440

SUBROUTINE IFOS(RI,RJ,RK,R) ORB06450
* THIS SUBROUTINE ASKS THE USER FOR THE INITIAL POSITION OF THE ORB06460
* Satellite IN GEOCENTRIC-EQUATORIAL COORDINATE SYSTEM ORB06470
ORB06480
ORB06490
ORB06500

DOUBLE PRECISION RI,RJ,RK,R ORB06510
ORB06520

CHARACTER*1, CHOICE ORB06530
LOGICAL CORREC ORB06540
CORREC = .FALSE. ORB06550
ORB06560

* PROMPT USER FOR VELOCITY VECTOR ORB06570
180 IF(.NOT.CORREC) THEN ORB06580
  CALL EXCMS('CLRSCRN') ORB06590
  PRINT*, 'ENTER RADIUS VECTOR VALUES IN "KM"' ORB06600
  PRINT*, 'RADIUS OF THE EARTH = 6400 KM' ORB06610
  CORREC = .TRUE. ORB06620
  PRINT*, 'ENTER RI :' ORB06630
  READ*, RI ORB06640
  PRINT*, 'RI = ', RI, 'KM' ORB06650
  PRINT*, 'ENTER RJ :' ORB06660

```

```

READ*, RJ
PRINT*, 'RJ = ', RJ, 'KM'
PRINT*, 'ENTER RK : '
READ*, RK
PRINT*, 'RK = ', RK, 'KM'

*      CALCULATE TOTAL R
R = DSQRT((RI**2) + (RJ**2) + (RK**2))
PRINT*, 'R = ', R, 'KM'
IF (R .LE. 6400.0) THEN
    PRINT*, 'RADIUS TO SMALL!! ENTER NEW VALUES!!'
    GOTO 180
ENDIF

*      CHECK WITH USER THAT Values ARE CORRECT
PRINT*, 'ARE THESE VALUES CORRECT?'
PRINT*, 'ENTER "Y" OR "N" : '
READ*, CHOICE
CHOICE = 'Y'
PRINT*, CHOICE
IF (CHOICE .EQ. 'Y') THEN
    CORREC = .TRUE.
ENDIF
GOTO 180
ENDIF
RETURN
END

*****SUBROUTINE IVEL(VI,VJ,VK,V,R,MU)
* THIS SUBROUTINE ASKS THE USER FOR THE INITIAL VELOCITY OF THE
* Satellite

DOUBLE PRECISION VI,VJ,VK,V,R,VCIR,VMAX,MU

CHARACTER*1, CHOICE
LOGICAL CORREC
CORREC = .FALSE.

*      CALCULATE ESCAPE VELOCITY AND CIRCULAR VELOCITY AND PROMPT USER
*      FOR VELOCITY VECTOR
190 IF(.NOT.CORREC) THEN
    CALL EXCMS('CLRSCRN')
    VCIR = DSQRT(MU/R)
    VMAX = DSQRT((2.0*MU)/R)
    PRINT*, 'CIRCULAR VELOCITY = ', VCIR, 'KM/SEC'
    PRINT*, 'MAXIMUM VELOCITY = ', VMAX, 'KM/SEC'
    CORREC = .TRUE.
    PRINT*, 'ENTER VELOCITY VECTOR IN (KM/SEC)'

    PRINT*, 'ENTER VI : '
    READ*, VI
    PRINT*, 'VI = ', VI, 'KM/SEC'
    PRINT*, 'ENTER VJ : '
    READ*, VJ

```

35

```

PRINT*, 'VJ = ', VJ, 'KM/SEC' ORB07230
PRINT*, 'ENTER VK : ' ORB07240
READ*, VK ORB07250
PRINT*, 'VK = ', VK, 'KM/SEC' ORB07260
ORB07270
* CALCULATE TOTAL VELOCITY (V) ORB07280
V = DSQRT((VI**2) + (VJ**2) + (VK**2)) ORB07290
PRINT*, 'V = ', V, 'KM/SEC' ORB07300
ORB07310
* CHECK WITH USER THAT VALUES ARE CORRECTS ORB07320
PRINT*, 'ARE THESE VALUES CORRECT?' ORB07330
PRINT*, ' ENTER "Y" OR "N" : ' ORB07340
READ*, CHOICE ORB07350
CHOICE = 'Y' ORB07360
PRINT*, CHOICE ORB07370
IF (CHOICE.EQ.'Y') THEN ORB07380
    CORREC = .TRUE. ORB07390
ENDIF ORB07400
IF (V.GE.VMAX) THEN ORB07410
    PRINT*, 'VELOCITY IS GREATER THAN THE ESCAPE VELOCITY!!' ORB07420
    PRINT*, 'RE-ENTER VELOCITY!!!' ORB07430
    CORREC = .FALSE. ORB07440
ENDIF ORB07450
GOTO 190 ORB07460
ENDIF ORB07470
RETURN ORB07480
END ORB07490
***** ORB07500
SUBROUTINE INELTS(RI, RJ, RK, R, VI, VJ, VK, V, MU, PI) ORB07510
* SATELLITE PLACED ON 'I' AXIS AND USER SUPPLY ORBITAL ELEMENTS TO ORB07520
* GET INITIAL POSITION AND VELOCITY ORB07530
ORB07540
ORB07550
ORB07560
DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,MU,I,ENR,A,E,RP,RA,PI,VMAX ORB07570
CHARACTER*1,CHOICE ORB07580
ORB07590
* PROMPT USER FOR PERIGEE RADIUS ORB07600
198 PRINT*, 'ENTER RADIUS OF PERIGEE(RP) IN (KM), FOR EXAMPLE:' ORB07610
PRINT*, 'LOW EARTH ORBIT (LEO), RP = 6600.0 KM' ORB07620
PRINT*, 'GEOSYNCHROUNOUS ORBIT, RP = 42241.1 KM' ORB07630
PRINT*, 'ENTER RP:' ORB07640
PRINT*, '"RP" MUST BE > 6400KM' ORB07650
READ*, RP ORB07660
PRINT*, RP ORB07670
ORB07680
* CHECK FOR VALID RADIUS ORB07690
IF (RP.LT.6400.0) THEN ORB07700
    PRINT*, 'YOUR "RP" IS TO SMALL!!' ORB07710
    GOTO 198 ORB07720
ENDIF ORB07730
ORB07740
* PROMPT USER FOR TYPE OF INPUT ORB07750
PRINT*, 'DO YOU WANT TO ENTER THE ECCENTRICITY (E), ' ORB07760
PRINT*, 'RADIUS OF APOGEE (RA), OR VELOCITY (V)?' ORB07770
PRINT*, 'ENTER "E", "R", OR "V": ' ORB07780

```

```

READ*,CHOICE          ORB07790
PRINT*,CHOICE         ORB07800
CALL ENCLS('CLRSRN') ORB07810
ORB07820

* USER ENTERS Eccentricity AND SEMI-MAJOR AXIS, ENERGY AND VELOCITY ORB07830
* IS CALCULATED IN THAT ORDER ORB07840
IF (CHOICE .EQ. 'E') THEN ORB07850
  PRINT*, 'ENTER ECCENTRICITY (E):' ORB07860
  PRINT*, '0.0 <= E < 1.0' ORB07870
  READ*,E               ORB07880
  PRINT*,E              ORB07890
  ORB07900

★ CHECK FOR VALID ECCENTRICITY ORB07910
  IF ((E .LT. 0.0) .OR. (E .GE. 1.0)) THEN ORB07920
    PRINT*, 'INVALID "E"' ORB07930
    GOTO 198              ORB07940
  ENDIF                  ORB07950
  A = RP/(1-E)           ORB07960
  ENR = -MU/(2.0*A)      ORB07970
  V = DSQRT(2*(ENR+(MU/RP))) ORB07980
  ORB07990

* USER INPUTS RADIUS OF APOGEE AND ECCENTRICITY IS CALCULATED ORB08000
* THEN SEMI-MAJOR AXIS, ENERGY AND THEN VELOCITY. ORB08010
ELSEIF (CHOICE .EQ. 'R') THEN ORB08020
  PRINT*, 'ENTER RADIUS OF APOGEE (RA) IN KM:' ORB08030
  PRINT*, '"RA" MUST BE >="RP", "RP" = ',RP ORB08040
  READ*,RA               ORB08050
  PRINT*,RA              ORB08060
  ORB08070

* CHECK FOR VALID RADIUS OF APOGEE ORB08080
  IF (RA .LT. RP) THEN ORB08090
    PRINT*, 'YOUR "RA" IS TO SMALL!' ORB08100
    GOTO 198              ORB08110
  ENDIF                  ORB08120
  E = (RA-RP)/(RA+RP)     ORB08130
  A = RP/(1-E)            ORB08140
  ENR = -MU/(2.0*A)       ORB08150
  V = DSQRT(2*(ENR+(MU/RP))) ORB08160
  ORB08170

* USER INPUTS MAGNITUDE OF VELOCITY, PROGRAM PROVIDES CIRCULAR ORB08180
* AND ESCAPE VELOCITY FOR COMPARISON AND TO CHECK FOR VALID ORB08190
* INPUTS ORB08200
ELSEIF (CHOICE .EQ. 'V') THEN ORB08210
  PRINT*, 'ENTER VELOCITY IN KM/SEC:' ORB08220
  PRINT*, 'THE MINIMUM VELOCITY ALLOWED IS FOR A CIRCULAR ORBIT' ORB08230
  VCIRC = SQRT(SNGL(MU/RP)) ORB08240
  PRINT*, 'ORBIT. V(Circular) = ',VCIRC,' KM/S' ORB08250
  VMAX = DSQRT(2*(MU/RP)) ORB08260
  PRINT*, 'THE MAXIMUM VELOCITY < ',VMAX,' KM/S' ORB08270
  READ*,V                 ORB08280
  PRINT*,V                ORB08290
  IF (V .LT. VCIRC) THEN ORB08300
    PRINT*, 'VELOCITY TO SMALL!' ORB08310
    GOTO 198              ORB08320
  ENDIF                  ORB08330
  IF (V .GE. VMAX) THEN ORB08340

```

```

        PRINT*, 'VELOCITY TO GREAT!!'
        GOTO 198
    ENDIF
ELSE
    PRINT*, 'INVALID ENTRY! TRY AGAIN'
    GOTO 198
ENDIF

* INCLINATION NEEDED TO GIVE Velocity A Direction
PRINT*, 'ENTER INCLINATION (I) IN DEGREES:'
READ*, I
PRINT*, I
I = (PI/180.0)*I
VK = V*DSIN(I)
VJ = V*DCOS(I)
VI = 0.0

* RADIUS VECTOR SET
RI = RP
RJ = 0.0
RK = 0.0
R = RP
RETURN
END

*****+
* CALCULATE THE ORBITAL ELEMENTS
*****+
SUBROUTINE CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,
+                   LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)
* THIS SUBROUTINE CALLS THE INDIVIDUAL SUBROUTINES TO CALCULATE THE
* ORBITAL ELEMENTS
* THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES(RETURNED VALUES)
* ENERGY = ENERGY PER MASS (ENR)
* ANGMOM = ANGULAR MOMENTUM (H,HI,HJ,HK)
* NODE = NODE VECTOR (N,NI,NJ,NK)
* LATREC = SEMI-LATUS RECTUS (P)
* ECC = ECCENTRICITY (E,EI,EJ,EK)
* SMAXIS = SEMI-MAJOR AXIS (A)
* INCL = INCLINATION (I)
* ASNODE = LONGITUDE OF ASCENDING NODE (LAN)
* ARP = ARGUMENT OF PERIGEE (AP)
* IJKPQW = 'IJK' SYSTEM TO 'PQW' SYSTEM
* TANOM = TRUE ANOMALY (TA)
* ARLAT = ARGUMENT OF LATITUDE (AL)
* LONPER = LONGITUDE OF Perigee (LP)
* TLON = TRUE LONGITUDE (TL)
* PERIOD = PERIOD (PER)
* ECCAN = ECCENTRIC ANOMALY (EA)
* MEANMO = MEAN MOTION (MM)
* MEANAN = MEAN ANOMALY (MA)
* TFLGHT = TIME OF FLIGHT (TF)

DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,AL,

```

```

+ LP,TA,PER,EA,MA,AP,TF,HI,HJ,H,NI,NJ,NK,N,P,PI,MU,MM,ENR, ORB08910
+ TL,RP,RQ,RW,NP,NQ,NW ORB08920
CALL ENERGY(V,R,MU,ENR) ORB08930
CALL ANGMOM(RI,RJ,RK,VI,VJ,VK,HI,HJ,HK,H) ORB08940
CALL NODE(HI,HJ,NI,NJ,NK,N) ORB08950
CALL LATREC(H,P,MU) ORB08960
CALL ECC(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,MU) ORB08970
CALL SMANIS(MU,ENR,A) ORB08980
CALL INCL(HK,H,I,PI) ORB08990
CALL ASNODE(NI,N,LAN,NJ,PI) ORB09000
CALL ARP(NI,NJ,N,EI,EJ,EK,E,AP,PI,NP,NQ,LAN) ORB09010
ELSE ORB09020
  LAN = 0.0 ORB09030
  AP = 0.0 ORB09040
ENDIF ORB09050
ORB09060
ORB09070
ORB09080
ORB09090
ORB09100
* COORDINATE TRANSFORMATION OF 'R' AND 'V' VECTORS ORB09110
CALL IJKPOW(LAN,AP,I,RI,RJ,RK,RP,RQ,RW) ORB09120
CALL IJKPOW(LAN,AP,I,NI,NJ,NK,NP,NQ,NW) ORB09130
CALL TANON(EI,EJ,EK,E,RI,RJ,RK,RP,RQ,RW,R,VI,VJ,VK,TA,PI) ORB09140
ORB09150
* SPECIAL CASE FOR Inclination = 0.0 ORB09160
IF (I .NE. 0.0) THEN ORB09170
  CALL ARLAT(NI,NJ,NK,N,RI,RJ,RK,R,AL,PI,TA,AP) ORB09180
ELSE ORB09190
  AL = TA ORB09200
ENDIF ORB09210
CALL LONPER(LAN,AP,LP) ORB09220
CALL TLON(LAN,AP,TA,TL) ORB09230
CALL PERIOD(A,PER,PI,MU) ORB09240
CALL ECCANE(TA,EA,PI) ORB09250
CALL MEANMO(A,MM,MU) ORB09260
CALL MEANAN(EA,E,MA) ORB09270
CALL TFLGHT(MM,MA,TF) ORB09280
RETURN ORB09290
END ORB09300
ORB09310
***** ORB09320
SUBROUTINE ENERGY(V,R,MU,ENR) ORB09330
* THIS SUBROUTINE CALCULATES THE ENERGY OF THE ORBIT ORB09340
DOUBLE PRECISION V,R,MU,ENR ORB09350
ORB09360
ENR = ((V**2)/2) - (MU/R) ORB09370
RETURN ORB09380
END ORB09390
ORB09400
ORB09410
ORB09420
ORB09430
ORB09440
ORB09450
ORB09460
* SUBROUTINE ANGMOM(RI,RJ,RK,VI,VJ,VK,HI,HJ,HK,H)
* THIS SUBROUTINE CALCULATES THE ANGULAR MOMENTUM

```

```

DOUBLE PRECISION RI,RJ,RK,VI,VJ,VK,HI,HJ,NK,H          ORB09470
HI = (RJ * VK) - (RK * VJ)                            ORB09480
HJ = (RK * VI) - (RI * VK)                            ORB09490
HK = (RI * VJ) - (RJ * VI)                            ORB09500
H = DSQRT((HI**2) + (HJ**2) + (HK**2))              ORB09510
RETURN                                                 ORB09520
END                                                   ORB09530
*****                                                 ORB09540
SUBROUTINE NODE(HI,HJ,NI,NJ,NK,N)                      ORB09550
* THIS SUBROUTINE CALCULATES THE NODE VECTOR           ORB09560
DOUBLE PRECISION HI,HJ,NI,NJ,NK,N                      ORB09570
NI = -HJ                                              ORB09580
NJ = HI                                              ORB09590
NK = 0.0                                              ORB09600
N = DSQRT((NI**2) + (NJ**2))                          ORB09610
RETURN                                                 ORB09620
END                                                   ORB09630
*****                                                 ORB09640
SUBROUTINE LATREC(H,P,MU)                             ORB09650
* THIS SUBROUTINE CALCULATES THE SEMI-LATUS RECTUM    ORB09660
DOUBLE PRECISION H,P,MU                               ORB09670
P = (H**2)/MU                                         ORB09680
RETURN                                                 ORB09690
END                                                   ORB09700
*****                                                 ORB09710
SUBROUTINE ECC(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,MU) ORB09720
* THIS SUBROUTINE CALCULATES THE ECCENTRICITY          ORB09730
DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,MU,DOT ORB09740
CALCULATE DOT PRODUCT OF 'R' AND 'V' VECTORS          ORB09750
DOT = (RI*VI) + (RJ*VJ) + (RK*VK)                   ORB09760
EI = (1.0D+00/MU) * (((V**2) - (MU/R)) * RI - (DOT)*VI) ORB09770
EJ = (1.0D+00/MU) * (((V**2) - (MU/R)) * RJ - (DOT)*VJ) ORB09780
EK = (1.0D+00/MU) * (((V**2) - (MU/R)) * RK - (DOT)*VK) ORB09790
E = DSQRT((EI**2) + (EJ**2) + (EK**2))            ORB09800
RETURN                                                 ORB09810
END                                                   ORB09820
*****                                                 ORB09830
SUBROUTINE SMAXIS(MU,ENR,A)                           ORB09840
* THIS SUBROUTINE Calculates THE SEMI-MAJOR AXIS       ORB09850
ORB10000                                             ORB09860
ORB10010                                             ORB09870
ORB10020                                             ORB09880

```

```

DOUBLE PRECISION MU,ENR,A          ORB10030
A = -MU/(2*ENR)                   ORB10040
RETURN                            ORB10050
END                               ORB10060
*****                                ORB10070
                                     ORB10080
                                     ORB10090
                                     ORB10100
SUBROUTINE INCL(HK,H,I,PI)         ORB10110
* THIS SUBROUTINE CALCULATES THE INCLINATION   ORB10120
* 'I' ALWAYS LESS THAN 180 DEGREES             ORB10130
                                         ORB10140
DOUBLE PRECISION HK,H,I,PI        ORB10150
I = DACOS(HK/H)                  ORB10160
RETURN                            ORB10170
END                               ORB10180
ORB10190
ORB10200
*****                                ORB10210
                                     ORB10220
SUBROUTINE ASNODE(NI,N,LAN,NJ,PI)  ORB10230
* THIS SUBROUTINE CALCULATES THE LONGITUDE OF THE ASCENDING NODE   ORB10240
* IF 'NJ' > 0      THEN 'LAN' < 180 DEGREES             ORB10250
                                         ORB10260
DOUBLE PRECISION NI,N,LAN,NJ,PI   ORB10270
LAN = DATAN2(NJ,NI)              ORB10280
IF (LAN .LT. 0.0) THEN           ORB10290
    LAN = (2*PI) + LAN
ENDIF                            ORB10300
RETURN                            ORB10310
END                               ORB10320
ORB10330
ORB10340
ORB10350
ORB10360
ORB10370
*****                                ORB10380
SUBROUTINE ARP(NI,NJ,N,EI,EJ,EK,E,AP,PI,NP,NQ,LAN)   ORB10390
* THIS SUBROUTINE CALCULATES THE ARGUMENT OF Perigee   ORB10400
* IF 'EK' GREATER THAN 0 THEN 'AP' < 180               ORB10410
* VARIABLE TEMP USED AS A Temporary VALUE FOR ARCTAN   ORB10420
                                         ORB10430
DOUBLE PRECISION NI,NJ,N,EI,EJ,EK,E,AP,PI,NQ,NP,TEMP,LAN  ORB10440
IF ((EI .EQ. 0.0) .AND. (EJ .EQ. 0.0)) THEN           ORB10450
    AP = 0.0
ELSE
    TEMP = DATAN2(EJ,EI)
    IF (TEMP .GT. LAN) THEN                         ORB10460
        AP = TEMP - LAN
    ELSE
        AP = (2*PI) - (LAN - TEMP)                 ORB10470
    ENDIF
    IF (AP .LT. 0.0) THEN                           ORB10480
        AP = (2*PI) + AP
    ENDIF
    IF (AP .GT. (2*PI)) THEN                      ORB10490
        AP = AP - (2*PI)
    ENDIF

```

```
      ENDIF  
      ENDIF  
      RETURN  
      END
```

ORB10590  
ORB10600  
ORB10610  
ORB10620  
ORB10630  
ORB10640  
ORB10650  
ORB10660  
ORB10670  
ORB10680  
ORB10690  
ORB10700  
ORB10710  
ORB10720  
ORB10730  
ORB10740  
ORB10750  
ORB10760  
ORB10770  
ORB10780  
ORB10790  
ORB10800

```
*****  
SUBROUTINE TANOM(EI,EJ,EK,E,RI,RJ,RK,RP,RQ,RW,R,VI,VJ,VK,  
+ TA,PI)  
* THIS SUBROUTINE CALCULATES THE TRUE Anomaly  
* IF (R DOT V) > 0 THEN TA < 180 DEGREES  
  
DOUBLE PRECISION DOT,EI,EJ,EK,E,RI,RJ,RK,R,VI,VJ,VK,TA,PI,  
+ RP,RQ,RW  
  
TA = DATAN2(RQ,RP)  
IF (TA . LT. 0.0 ) THEN  
    TA = (2 * PI) + TA  
ENDIF  
RETURN  
END
```

ORB10810  
ORB10820  
ORB10830  
ORB10840  
ORB10850  
ORB10860  
ORB10870  
ORB10880  
ORB10890  
ORB10900  
ORB10910  
ORB10920

```
*****  
SUBROUTINE ARLAT(NI,NJ,NK,N,RI,RJ,RK,R,AL,PI,TA,AP)  
* THIS SUBROUTINE CALCULATES THE ARGUMENT OF LATITUDE  
* IF (RK > 0) THEN AL < 180 DEGREES  
  
DOUBLE PRECISION NI,NJ,NK,N,RI,RJ,RK,R,AL,PI,TA,AP  
  
AL = TA + AP  
RETURN  
END
```

ORB10930  
ORB10940  
ORB10950  
ORB10960  
ORB10970  
ORB10980  
ORB10990  
ORB11000  
ORB11010  
ORB11020  
ORB11030

```
*****  
SUBROUTINE LONPER(LAN,AP,LP)  
* THIS SUBROUTINE CALCULATES THE LONGITUDE OF PERIGEE  
  
DOUBLE PRECISION LAN,AP,LP  
  
LP = LAN + AP  
RETURN  
END
```

ORB11040  
ORB11050  
ORB11060  
ORB11070  
ORB11080  
ORB11090  
ORB11100  
ORB11110  
ORB11120  
ORB11130

```
*****  
SUBROUTINE TLON(LAN,AP,TA,TL)  
* THIS SUBROUTINE CALCULATES THE TRUE LONGITUDE AT EPOCH  
  
DOUBLE PRECISION LAN,AP,TA,TL  
  
TL = AP + LAN + TA  
RETURN  
END
```

```

***** SUBROUTINE PERIOD(A,PER,PI,MU) ***** ORB11140
* THIS SUBROUTINE CALCULATES THE PERIOD ORB11150
DOUBLE PRECISION A,PER,PI,MU ORB11160
PER = 2.0D+00*(PI)*DSQRT((A**3)/MU) ORB11170
RETURN ORB11180
END ORB11190
***** ORB11200
***** SUBROUTINE ECCAN(E,TA,EA,PI) ***** CRB11210
* THIS SUBROUTINE CALCULATES THE ECCENTRIC Anomaly ORB11220
DOUBLE PRECISION E,TA,EA,PI ORB11230
EA = DACOS((E + DCOS(TA))/(1.0D+00 + E*DCOS(TA))) ORB11240
IF (TA .GT. PI) THEN ORB11250
  EA = (2*PI) - EA ORB11260
ENDIF ORB11270
RETURN ORB11280
END ORB11290
***** ORB11300
***** SUBROUTINE MEANMO(A,MM,MU) ***** ORB11310
* THIS SUBROUTINE CALCULATES THE MEAN MOTION ORB11320
DOUBLE PRECISION A,MM,MU ORB11330
MM = DSQRT(MU/(A**3)) ORB11340
RETURN ORB11350
END ORB11360
***** ORB11370
***** SUBROUTINE MEANAN(EA,E,MA) ***** ORB11380
* THIS SUBROUTINE CALCULATES THE MEAN Anomaly ORB11390
DOUBLE PRECISION EA,E,MA ORB11400
MA = EA - E*DSIN(EA) ORB11410
RETURN ORB11420
END ORB11430
***** ORB11440
***** SUBROUTINE TFLGHT(MM,MA,TF) ***** ORB11450
* THIS SUBROUTINE CALCULATES THE TIME OF FLIGHT ORB11460
DOUBLE PRECISION MM,MA,TF ORB11470
TF = (1/MM)**MA ORB11480

```

```

RETURN ORB11700
END ORB11710
***** ORB11720
* CALCULATE UNPERTURBED ORBIT ORB11730
***** ORB11740
***** ORB11750
***** ORB11760
SUBROUTINE UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R, ORB11770
+ VI,VJ,VK,V,MU,PI,H,A,E,N,TA,P,MM,MA,EA, ORB11780
+ TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY, ORB11790
+ TT) ORB11800
* THIS SUBROUTINE CALCULATE THE UNPERTURBED ORBIT ORB11810
* THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES: ORB11820
* NEWELT = CALCULATE NEW ELEMENTS AFTER TIME STEP ORB11830
* NEWPOS = CALCULATE NEW POSITION AFTER TIME STEP ORB11840
* NEWVEL = CALCULATE NEW VELOCITY AFTER TIME STEP ORB11850
* STORE = STORES POSITION IN ARRAYS ORB11860
STORE = STORES POSITION IN ARRAYS ORB11870
DOUBLE PRECISION T,DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, ORB11880
+ MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,TT ORB11890
DIMENSION RARAY(500),TARAY(500),RIRAY(500),RJRAY(500), ORB11900
+ RKRAY(500),AINRAY(500),APRAY(500),TIMRAY(500) ORB11910
* SET TRUE ANOMALY TO NEGATIVE SO LOOP CAN BE EXECUTED ORB11920
IF (TA .GT. 6.21) THEN ORB11930
  TA = TA - (2*PI) ORB11940
ENDIF ORB11950
CONTINUE AROUND ORBIT TILL CLOSE TO PERIGEE ORB12000
230 IF ((TA .LE. 6.21) .AND. (T .LE. PER)) THEN ORB12010
  ORB12020
* Increment TRUE TIME ORB12030
  TT = TT + DT ORB12040
  CALL NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER) ORB12050
  CALL NPOS(RI,RJ,RK,R,LAN,AP,I,TA,A,E) ORB12060
  CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU) ORB12070
  ORB12080
* INCREMENT STEP COUNTER AND STORE VALUES ORB12090
  NUM = NUM + 1 ORB12100
  CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY, ORB12110
+ RARAY,TARAY,NUM,I,AP,AINRAY,APRAY, ORB12120
+ TT,TIMRAY) ORB12130
  ORB12140
* INCREMENT TIME STEP COUNTER ORB12150
  T = T + DT ORB12160
  GOTO 230 ORB12170
ENDIF ORB12180
RETURN ORB12190
END ORB12200
***** ORB12210
* CALCULATE THE UNPERTURBED NEW ELEMENTS ORB12220
***** ORB12230
***** ORB12240
***** ORB12250

```

```

SUBROUTINE NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER) ORB12260
* THIS SUBROUTINE CALCULATES THE Unperturbed NEW ELEMENTS ORB12270
ORB12280
* THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES: ORB12290
* NEA = NEW ECCENTRIC ANOMALY ORB12300
* NTA = NEW TRUE ANOMALY ORB12310
ORB12320
DOUBLE PRECISION MM,MA,E,EA,TA,TF,DT,PI,PER ORB12330
ORB12340
* Increment TIME OF FLIGHT AND CHECK IF TF GREATER THAN PERIOD ORB12350
TF = TF + DT ORB12360
IF (TF .GT. PER) THEN ORB12370
  TF = TF - PER ORB12380
ENDIF ORB12390
ORB12400
* CALCULATE MEAN ANOMALY AND USE TO FIND ECCENTRIC Anomaly THEN NEW ORB12410
* TRUE ANOMALY ORB12420
MA = MM*(TF) ORB12430
CALL NEA(MA,E,EA) ORB12440
CALL NTA(EA,E,TA,PI) ORB12450
RETURN ORB12460
END ORB12470
ORB12480
***** ORB12490
* CALCULATE PERTURBED ORBIT ORB12500
***** ORB12510
ORB12520
SUBROUTINE PRETUR(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R, ORB12530
+ VI,VJ,VK,V,FR,FS,MU,PI,H,A,E,N,TA,P,MM,MA,EA, ORB12540
+ TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY, ORB12550
+ TT,TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDMY,TDMA,TDLAN,TDH,TDAP) ORB12560
* THIS SUBROUTINE CALCULATES THE PERTURBED ORBIT. ORB12570
ORB12580
* THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES: ORB12590
* TFORCE = CALCULATE THE TOTAL PERTURBING FORCE ON THE SATELLITE ORB12600
* PNNEWEL = CALCULATE THE Perturbed NEW ELEMENTS ORB12610
* NPOS = NEW POSITION AFTER TIME STEP ORB12620
* NVEL = NEW VELOCITY AFTER TIME STEP ORB12630
* PERIOD = PERIOD OF PERTURBED ORBIT ORB12640
* STORE = STORE POSITION AND ELEMENTS IN ARRAYS FOR PLOTTING ORB12650
ORB12660
DOUBLE PRECISION T,DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, ORB12670
+ FR,FS,FW,MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,TT, ORB12680
+ DI,DA,DE,DMM,DMA,DLAN,DH,DAP,EI,EJ,EK,HI,HJ,LP,M, ORB12690
+ DVR,DVS,DVW,DVJ,DVK ORB12700
ORB12710
DIMENSION RARAY(500),TARAY(500),RIRAY(500),RJRAY(500), ORB12720
+ RKRAY(500),AINRAY(500),APRAY(500),TIMRAY(500) ORB12730
ORB12740
* SET MEAN RADIUS OF EARTH ORB12750
RE = 6400.0 ORB12760
ORB12770
DT = PER/50 ORB12780
T = DT ORB12790
IF (TA .GT. 6.21) THEN ORB12800
  TA = TA - (2*PI) ORB12810

```

```

ENDIF ORB12820
IF (TF .GE. PER) THEN ORB12830
  TF = TF - PER CRB12840
ENDIF ORB12850
CRB12860

* CONTINUE Around ORBIT FOR ONE PERIOD ORB12870
240 IF ((TF .LT. PER) .AND. (T .LT. PER)) THEN ORB12880
  ORB12890

* INCREMENT TRUE TIME ORB12900
  TT = TT + DT ORB12910
  CALL TFORCE(AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, ORB12920
    + TT,FR,FS,FW,MU,PI, ORB12930
    + FEA,FSU,FMO,FDRA,FOR, ORB12940
    + EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,TF,P, ORB12950
    + MM,N,H,HI,HJ,DT) ORB12960
  CALL PNEWEL(FR,FS,FW,H,R,A,E,T,DT,I,LAN,AL, ORB12970
    + APP,MM,MA,EA,TF,T,MU,PI, ORB12980
    + DI,DA,DE,DMM,DMA,DLAN,DH,DAP) ORB12990
  CALL NPOS(RI,RJ,RK,R,LAN,AP,I,TA,A,E) ORB13000
  CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU) ORB13010
  ORB13020

* CALCULATE NEW PERIOD AND RESET TIME STEP AND TIME COUNTER ORB13030
* IF NOT AT END OF ORBIT ORB13040
  IF (T .LY. (PER-DT)) THEN ORB13050
    CALL PERIOD(A,PER,PI,MU) ORB13060
    DT = PER/50 ORB13070
    T = TF ORB13080
  ENDIF ORB13090
  ORB13100

* INCREMENT STEP COUNTER ORB13110
  NUM = NUM + 1 ORB13120
241 CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY, ORB13130
  + RARAY,TARAY,NUM,I,AP,AINRAY,APRAY, ORB13140
  + TT,TIMRAY) ORB13150
  ORB13160

* TOTAL ELEMENT CHANGES ORB13170
  TDI = TDI + SNGL(ABS(DI)) ORB13180
  TDA = TDA + SNGL(ABS(DA)) ORB13190
  TDE = TDE + SNGL(ABS(DE)) ORB13200
  TDMM = TDMM + SNGL(ABS(DMM)) ORB13210
  TDMA = TDMA + SNGL(ABS(DMA)) ORB13220
  TDLAN = TDLAN + SNGL(ABS(DLAN)) ORB13230
  TDH = TDH + SNGL(ABS(DH)) ORB13240
  TDAP = TDAP + SNGL(ABS(DAP)) ORB13250
  TFEA = TFEA + FEA ORB13260
  TFSU = TFSU + FSU ORB13270
  TFMO = TFMO + FMO ORB13280
  TFDRA = TFDRA + FDRA ORB13290
  ORB13300

* CHECK FOR IMPACT ORB13310
  IF (R .LE. RE) THEN ORB13320
    PRINT*, 'SATELLITE WILL IMPACT THE EARTH!!' ORB13330
    T = PER ORB13340
  ENDIF ORB13350
  ORB13360
  CRB13370

* INCREMENT TIME COUNTER

```

```

T = T + DT          ORB13380
GOTO 240          ORB13390
ENDIF              ORB13400
RETURN             ORB13410
END                ORB13420
*****              ORB13430
* CALCULATE THE PERTURBING FORCES          ORB13440
*****              ORB13450
*****              ORB13460
*****              ORB13470
SUBROUTINE TFORCE(AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,TT,    ORB13480
+           FR,FS,FW,MU,PI,FEA,FSU,FMO,FDRA,FOR,    ORB13490
+           EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,TF,P,    ORB13500
+           MM,N,H,HI,HJ,DT)    ORB13510
* THIS SUBROUTINE SUMS ALL THE PERTURBING FORCES FOR THE TOTAL    ORB13520
* PERTURBING FORCE.    ORB13530
* THE FOLLOWING SUBROUTINES WERE CALLED:    ORB13540
* OBERT = OBLATENESS OF THE EARTH    ORB13550
* FSUN = GRAVITATIONAL Attraction OF THE SUN    ORB13560
* FMOON = GRAVITATIONAL Attraction OF THE MOON    ORB13570
* FDRA = DRAG FORCES    ORB13580
* FDRAG = DRAG FORCES    ORB13590
DOUBLE PRECISION FER,FES,FEW,FSR,FSS,FSW,FMR,FMS,FMW,MU,PI,    ORB13600
+   FDR,FDS,FDW,FR,FS,FW,RI,RJ,RK,R,AL,I,TT,LAN,AP,VI,VJ,VK,V,    ORB13610
+   EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,TF,P,    ORB13620
+   MM,N,H,HI,HJ,DT)    ORB13630
CALL OBEART(RI,RJ,RK,R,AL,I,FER,FES,FEW,MU)    ORB13640
CALL FSUN(TT,RI,RJ,RK,R,FSR,FSS,FSW,PI)    ORB13650
CALL FMOON(TT,RI,RJ,RK,R,FMR,FMS,FMW,PI)    ORB13660
CALL FDRA(RI,RJ,RK,R,VI,VJ,TT,LAN,AP,I,FDR,FDS,FDW,    ORB13670
+           EI,EJ,EK,E,AL,LP,TA,PER,EA,MA,AL,TF,P,PI,MU,    ORB13680
+           MM,N,H,HI,HJ,DT)    ORB13690
* SUM VECTOR FORCES    ORB13700
* FR = FER + FSR + FMR + FDR    ORB13710
* FS = FES + FSS + FMS + FDS    ORB13720
* FW = FEW + FSW + FMW + FDW    ORB13730
* CALCULATE TOTAL FORCE FROM EACH, AND TOTAL OF ALL    ORB13740
* FEA = SNGL(SQRT((FER**2)+(FES**2)+(FEW**2)))    ORB13750
* FSU = SNGL(SQRT((FSR**2)+(FSS**2)+(FSW**2)))    ORB13760
* FMO = SNGL(SQRT((FMR**2)+(FMS**2)+(FMW**2)))    ORB13770
* FDRA = SNGL(SQRT((FDR**2)+(FDS**2)+(FDW**2)))    ORB13780
* FOR = SNGL(SQRT((FR**2)+(FS**2)+(FW**2)))    ORB13790
RETURN            ORB13800
END                ORB13810
*****              ORB13820
* SUBROUTINE OBEART(RI,RJ,RK,R,AL,I,FER,FES,FEW,MU)    ORB13830
* THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO THE    ORB13840
* OBLIQUENESS OF THE EARTH.    ORB13850
*                                     ORB13860
*                                     ORB13870
*                                     ORB13880
*                                     ORB13890
*                                     ORB13900
*                                     ORB13910
*                                     ORB13920
*                                     ORB13930

```

```

DOUBLE PRECISION J2,RE,FER,FES,FEW,RI,RJ,RK,R,AL,I,MU,N      ORB13940
*          J2 = 1.082364D-03                                     ORB13950
*          RE = 6.3763E+03                                     ORB13960
*          FER = ((-3.0D+00**MU**J2**((RE**2)))/(2.0D+00*(R**4)))** ORB13970
*          + (1.0D+00 - (3.0D+00* ((DSIN(I))**2)** ((DSIN(AL))**2)))** ORB13980
*          FES = ((-3.0D+00**MU**J2**((RE**2)))/(R**4))** ORB13990
*          + (((DSIN(I))**2)** (DSIN(AL)))*(DCOS(AL)))          ORB14000
*          FEW = ((-3.0D+00**MU**J2**((RE**2)))/(R**4))** ORB14010
*          + (DSIN(I)**DCOS(I)*DSIN(AL))                         ORB14020
*          RETURN                                                 ORB14030
*          END                                                   ORB14040
*****                                                       ORB14050
*          SUBROUTINE FSUN(TT,RI,RJ,RK,R,FSR,FSS,FSW,PI)        ORB14060
*          THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO THE SUN ORB14070
*          THE FOLLOWING SUBROUTINES ARE CALLED:                  ORB14080
*          SUNPOS = SUNS POSITION ORBITING AROUND EARTH          ORB14090
*          HEVBOD = PERTURBING FORCE FROM A Heavenly BODY         ORB14100
*          DOUBLE PRECISION FSR,FSS,FSW,PI,                      ORB14110
*          + RSI,RSJ,RSK,SLAN,SI,SAL,SMU,TT,RI,RJ,RK,R,RS          ORB14120
*          SUNS GRAVITATIONAL PARAMETER                           ORB14130
*          SMU = 1.3271544D+11                                     ORB14140
*          CALL SUNPOS(TT,RSI,RSJ,RSK,RS,SLAN,SI,SAL,PI)          ORB14150
*          CALL HEVBOD(RI,RJ,RK,R,RSI,RSJ,RSK,RS,SLAN,SAL,SI,SMU,FSR,FSS,FSW) ORB14160
*          RETURN                                                 ORB14170
*          END                                                   ORB14180
*****                                                       ORB14190
*          SUBROUTINE FMOON(TT,RI,RJ,RK,R,FMR,FMS,FMW,PI)       ORB14200
*          THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO The MOON ORB14210
*          THE FOLLOWING SUBROUTINE ARE CALLED:                  ORB14220
*          MONPOS = MOONS POSITION ORBITING AROUND THE EARTH     ORB14230
*          HEVBOD = PERTURBING FORCE FROM A HEAVENLY BODY         ORB14240
*          DOUBLE PRECISION FMR,FMS,FMW,RMI,RMJ,RMK,MLAN,MI,MAL,MMU, ORB14250
*          + TT,RI,RJ,RK,R,RM,PI                                    ORB14260
*          MOONS GRAVITATIONAL PARAMETER                          ORB14270
*          MMU = 4.90287D+03                                     ORB14280
*          CALL MONPOS(TT,RMI,RMJ,RMK,RM,MLAN,MI,MAL,PI)          ORB14290
*          CALL HEVBOD(RI,RJ,RK,R,RMI,RMJ,RNK,RM,MLAN,MAL,MI,MMU,FMR,FMS,FMW) ORB14300
*          RETURN                                                 ORB14310
*          END                                                   ORB14320
*****                                                       ORB14330
*          SUBROUTINE HEVBOD(RI,RJ,RK,R,RPI,RPJ,RPK,RP,LAN,AL,INC,MUP, ORB14340
*                                ORB14350
*                                ORB14360
*                                ORB14370
*                                ORB14380
*                                ORB14390
*                                ORB14400
*                                ORB14410
*                                ORB14420
*                                ORB14430
*                                ORB14440
*                                ORB14450
*                                ORB14460
*                                ORB14470
*                                ORB14480
*                                ORB14490

```

```

+ FHR,FHS,FHW) ORB14500
* THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO A ORB14510
* HEAVENLY BODY. ORB14520
* ORB14530
* THE FOLLOWING SUBROUTINE WAS CALLED: ORB14540
* IJKRSW = 'IJK' SYSTEM TO THE 'RSW' SYSTEM ORB14550
* ORB14560
* DOUBLE PRECISION DOT,FHI,FHJ,FHK,RI,RJ,RK,R,RPI,RPJ,RPK,RP, ORB14570
+ LAN,AL,INC,MUP,I,J,K,IP,JP,KP,M,FHR,FHS,FHW ORB14580
* ORB14590
* CALCULATE UNIT VECTOR FOR SATELLITE AND PERTURBING BODIES POSITION ORB14600
* I = RI/R ORB14610
* J = RJ/R ORB14620
* K = RK/R ORB14630
* IP = RPI/RP ORB14640
* JP = RPJ/RP ORB14650
* KP = RPK/RP ORB14660
* ORB14670
* CALCULATE DOT PRODUCT OF UNIT VECTORS ORB14680
* DOT = (( I*IP )+( J*JP )+( K*KP )) ORB14690
* ORB14700
* CALCULATE FORCES IN THE 'IJK' SYSTEM ORB14710
* FHI = (MUP/(RP**2))*(K/RP)*(3.0D+00*DOT*(IP)-(I)) ORB14720
* FHJ = (MUP/(RP**2))*(R/RP)*(3.0D+00*DOT*(JP)-(J)) ORB14730
* FHK = (MUP/(RP**2))*(R/RP)*(3.0D+00*DOT*(KP)-(K)) ORB14740
* ORB14750
* Transform FORCES TO THE RSW SYSTEM ORB14760
* CALL IJKRSW(LAN,AL,INC,FHI,FHJ,FHK,FHR,FHS,FHW) ORB14770
* RETURN ORB14780
* END ORB14790
***** ORB14800
* ORB14810
* ORB14820
* SUBROUTINE SUNPOS(TT,RSI,RSJ,RSK,RS,SLAN,SI,SAL,PI) ORB14830
* THIS SUBROUTINE CALCULATES THE SUNS POSITION ORB14840
* ORB14850
* VARIABLES USED TO DESCRIBE THE SUNS ORBIT: ORB14860
* SI = SUNS INCLINATION ORB14870
* SLAN= SUNS Longitude OF ASCENDING NODE ORB14880
* SAP = SUNS ARGUMENT OF PERIGEE ORB14890
* RS = SUNS ORBITAL RADIUS ORB14900
* STA = SUNS TRUE ANOMALY ORB14910
* SAL = SUNS ARGUMENT OF LONGITUDE ORB14920
* ORB14930
* DOUBLE PRECISION SLAN,SI,SAL,RS,STA,SAP,TT,RSI,RSK, ORB14940
+ RSJ,RSP,RSQ,RSW,PI ORB14950
* ORB14960
* SI = 4.09279709D-01 ORB14970
* SLAN = 0.0D+00 ORB14980
* SAP = 0.0D+00 ORB14990
* RS = 1.4959965D+08 ORB15000
* STA = ((2.0*PI)/(365.0 * 86400.0) * TT) ORB15010
* SAL = STA + SAP ORB15020
* ORB15030
* CALCULATE SUNS POSITION IN 'PQW' SYSTEM ORB15040
* RSP = RS*DCOS(STA) . ORB15050

```

RSQ = RS*DSIN(STA)	ORB15060
RSW = 0.0D+00	ORB15070
* TRANSFORM POSITION TO 'IJK' SYSTEM	ORB15080
CALL PQWIJK(SLA, SAP, SI, RSP, RSQ, RSW, RSI, RSJ, RSK)	ORB15090
RETURN	ORB15100
END	ORB15110
*****	ORB15120
*****	ORB15130
SUBROUTINE MONPOS(TT, RMI, RMJ, RMK, RM, MLAN, MI, MAL, PI)	ORB15140
* THIS SUBROUTINE CALCULATES THE MOONS POSITION	ORB15150
* VARIABLES USED TO DESCRIBE THE SUNS ORBIT:	ORB15160
* MI = MOONS INCLINATION	ORB15170
* MLAN = MOONS Longitude OF ASCENDING NODE	ORB15180
* MAP = MOONS ARGUMENT OF PERIGEE	ORB15190
* RM = MOONS ORBITAL RADIUS	ORB15200
* MTA = MOONS TRUE ANOMALY	ORB15210
* MAL = MOONS ARGUMENT OF LONGITUDE	ORB15220
DOUBLE PRECISION    MLAN, MAL, RM, TM, MTA, RMP, RMQ, RMW,	ORB15230
+                RMI, RMJ, RMK, I, .., PI	ORB15240
MI = 4.99164166D-01	ORB15250
RM = 3.844D+05	ORB15260
MLAN = 0.0	ORB15270
MTA = ((2.0*PI)/(27.3 * 3600) * TT)	ORB15280
MAP = 0.0D+00	ORB15290
MAL = MTA	ORB15300
* CALCULATE MOON POSITION IN 'PQW' SYSTEM	ORB15310
RIP = RM*DCOS(MTA)	ORB15320
RIQ = RM*DSIN(MTA)	ORB15330
RMW = 0	ORB15340
* TRANSFORM POSITION TO 'IJK' SYSTEM	ORB15350
CALL PQWIJK(MLAN, MAP, MI, RMP, RMQ, RMW, RMI, RMJ, RMK)	ORB15360
RETURN	ORB15370
END	ORB15380
*****	ORB15390
*****	ORB15400
SUBROUTINE FDRAG(RI, RJ, RK, R, VI, VJ, VK, V, LAN, AP, I, FDR, FDS, FDW,	ORB15410
+                EI, EJ, EK, E, A, T, LP, TA, PER, EA, MA, AL, TF, P, PI, MU,	ORB15420
+                MM, N, H, HI, HJ, DT)	ORB15430
* THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO DRAG	ORB15440
* THE FOLLOWING VARIABLES ARE USED TO MODEL THE ATMOSPHERE:	ORB15450
* RE = RADIUS OF EARTH	ORB15460
* M = MASS OF SATELLITE	ORB15470
* AR = FRONTAL SURFACE AREA OF SATELLITE	ORB15480
* Z = ALTITUDE OF SATELLITE	ORB15490
* K = EXPONENTIAL DECAY FACTOR	ORB15500
* DENO = NORMAL DENSITY	ORB15510
* CD = COEFFICIENT OF DRAG	ORB15520
	ORB15530
	ORB15540
	ORB15550
	ORB15560
	ORB15570
	ORB15580
	ORB15590
	ORB15600
	ORB15610

DOUBLE PRECISION MAG,M,K,FDR,FDS,FDW,RE,AR,Z,DENO,CD,DEN.	ORB15620
+ FDJ,FDK,FDI,RI,RJ,RK,V1,VJ,VK,V,LAN,AP,I,R,	ORB15630
+ EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,AL,TF,P,PI,NU,	ORB15640
+ MM,N,H,HI,HJ,DT,DVR,DVS,DIW,DVI,DVJ,DVK	ORB15650
RE = 6.378145D+03	ORB15660
M = 1.0D+02	ORB15670
AR = 2.0D+01	ORB15680
Z = R - RE	ORB15690
* DEPENDING ON ALTITUDE SET ATMOSPHERE VARIABLES	ORB15700
IF (Z.LE.1.5D+02) THEN	ORB15710
K = 4.74D-02	ORB15720
DENO = 1.225D+00	ORB15730
CD = 1.0D+00	ORB15740
ELSEIF (Z.LE.5.5D+02) THEN	ORB15750
K = 3.4614D-02	ORB15760
DENO = 1.79846D-01	ORB15770
CD = 2.0D+00	ORB15780
ELSE	ORB15790
K = 2.21698D-3	ORB15800
DENO = 1.015484D-07	ORB15810
CD = 2.0D+00	ORB15820
ENDIF	ORB15830
* CALCULATE ATMOSPHERIC DENSITY	ORB15840
DEN = DENO * DEXP(-K**2)	ORB15850
* CALCULATE MAGNITUDE OF DRAG FORCE AND LIMIT IT TO 1.0E-20	ORB15860
MAG = -(0.5D+00)**CD*AR*DENO*V*(1.0D-03)/M	ORB15870
IF (ABS(MAG) .LT. 1.0D-20) THEN	ORB15880
MAG = -1.0D-20	ORB15890
ENDIF	ORB15900
* GIVE DRAG FORCE A Direction OF MINUS THE VELOCITY	ORB15910
FDR = 0.0	ORB15920
FDS = MAG * V	ORB15930
FDW = 0.0	ORB15940
RETURN	ORB15950
END	ORB15960
*****	ORB15970
* CALCULATE PERTURBED NEW ELEMENTS	ORB15980
*****	ORB15990
SUBROUTINE PNEWEL(FR,FS,FW,H,R,A,E,N,TA,DT,I,LAN,AL,AP,P,	ORB16000
+ MM,MA,EA,TF,T,NU,PI,DI,DA,DE,DMM,DMA,DLAN,DII,DAP)	ORB16010
* THIS SUBROUTINE CALCULATES THE NEW ELEMENTS FROM THE PREVIOUS	ORB16020
* ELEMENTS ADDED TO THE RATES OF CHANGE FOR ONE STEP	ORB16030
* THE FOLLOWING SUBROUTINES ARE CALLED:	ORB16040
* RATE = CALCULATES RATES OF CHANGE OF ORBITAL ELEMENTS	ORB16050
* NANGMO = NEW ANGULAR MOMENTUM (NEWH)	ORB16060
* NSMA = NEW SEMI-MAJOR AXIS (NEWA)	ORB16070
* NECC = NEW ECCENTRICITY (NEWE)	ORB16080
	ORB16090
	ORB16100
	ORB16110
	ORB16120
	ORB16130
	ORB16140
	ORB16150
	ORB16160
	ORB16170

```

* NINCL = NEW INCLINATION (NEWI) ORB16180
* NASNOD = NEW LONGITUDE OF ASCENDING NODE (NEWLAN) ORB16190
* NARPER = NEW ARGUMENT OF PERIGEE ( NEWAP) ORB16200
* NMNMO = NEW MEAN MOTION (NEWMM) ORB16210
* MEANMO = MEAN MOTION (MM) ORB16220
* NMNAN = NEW MEAN ANOMALY (NEWMA) ORB16230
* NEA = NEW ECCENTRIC ANOMALY (EA) ORB16240
* NTA = NEW TRUE ANOMALY (TA) ORB16250
* TFLGHT = TIME OF FLIGHT (TF) ORB16260
* ORB16270
* DOUBLE PRECISION FR,FS,FW,DMM,H,R,A,E,N,TA,DT,I,LAN,AL,AP,P, ORB16280
+ MM,MA,EA,TF,T,MU,PI,DA,DH,DE,DI,DLAN,DAP,DMA, ORB16290
+ NEWH,NEWA,NEWE,NEWI,NEWLAN,NEWAP,NEWMM ORB16300
* INCREMENT TIME OF FLIGHT BY ONE TIME STEP AND CALCULATE RATES ORB16310
  TF = TF + DT ORB16320
  CALL RATES(DH,DA,DE,DI,DLAN,DAP,DMM,DMA,E,MM,R,A,FR,FS,FW, ORB16340
+ TA,AL,H,P,T,MU,I) ORB16350
* CALCULATE NEW ELEMENTS ORB16360
  CALL NANGMO(H,DT,DH,NEWH) ORB16370
  CALL NSMA(A,DT,DA,NEWA) ORB16380
  CALL NECC(E,DT,DE,NEWE) ORB16390
  CALL NINCL(I,DT,DI,NEWI) ORB16400
  CALL NASNOD(LAN,DT,DLAN,NEWLAN) ORB16410
  CALL NARPER(AP,DT,DAP,NEWAP) ORB16420
* SET ELEMENTS TO NEW ELEMENTS ORB16430
  A = NEWA ORB16440
  E = NEWE ORB16450
  I = NEWI ORB16460
  LAN = NEWLAN ORB16470
  AP = NEWAP ORB16480
  P = A * (1 - E**2) ORB16490
* MOVE THE SATELLITE ONE TIME STEP ORB16500
  CALL MEANMO(A,MM,MU) ORB16510
  CALL NMNAN(MA,MM,DT,TF,DMA,PI) ORB16520
  CALL NEA(MA,E,EA) ORB16530
  CALL NTA(EA,E,TA,PI) ORB16540
  CALL TFLGHT(MM,MA,TF) ORB16550
  AL = TA + AP ORB16560
  RETURN ORB16570
  END ORB16580
***** ORB16590
* CALCULATE THE RATES OF CHANGE OF THE ORBITAL Elements ORB16600
***** ORB16610
* SUBROUTINE RATES(DH,DA,DE,DI,DLAN,DAP,DMM,DMA,E,MM,R,A,FR,FS,FW, ORB16620
+ TA,AL,H,P,T,MU,I) ORB16630
* THIS SUBROUTINE Calls THE FOLLOWING SUBROUTINES TO CALCULATE THE ORB16640
* TIME RATE-OF- CHANGE OF THE ORBITAL ELEMENTS: ORB16650
* RSMAX = RATE-OF-CHANGE OF THE SEMI-MAJOR AXIS (DA) ORB16660
* RECC = RATE-OF-CHANGE OF THE ECCENTRICITY (DE) ORB16670
* RINC = RATE-OF-CHANGE OF THE INCLINATION (DI) ORB16680

```

```

* RLAN = RATE-OF-CHANGE OF THE Longitude OF THE ASCENDING NODE      ORB16740
* (DLAN)                                                               ORB16750
* RAP = RATE-OF-CHANGE OF THE ARGUMENT OF PERIGEE (DAP)             ORB16760
* RMM = RATE-OF-Change OF THE MEAN MOTION (DMM)                      ORB16770
* RMA = RATE-OF-CHANGE OF THE MEAN ANOMALY (DMA)                     ORB16780
* RANGMO = RATE-OF-CHANGE OF THE ANGULAR MOMENTUM (DH)               ORB16790
* DOUBLE PRECISION DH,DA,DE,DI,DLAN,DAP,DMM,DMA,E,MM,R,A,FR,FS,FW,   ORB16800
+ TA,AL,H,P,T,MU,I

CALL RSMAX(E,MM,R,A,FR,FS,DA,TA)                                     ORB16810
CALL RECC(E,MM,R,A,FR,FS,TA,DE)                                       ORB16820
CALL RINC(E,MM,R,A,FW,AL,DI)                                         ORB16830
CALL RLAN(E,MM,R,A,I,FW,AL,DLAN)                                       ORB16840
CALL RAP(E,MM,R,A,I,H,P,AL,TA,FR,FS,FW,DAP)                         ORB16850
CALL RMM(MM,A,DMM,DA,MU)                                              ORB16860
CALL RMA(E,MM,R,A,TA,DMM,FR,FS,DMA,T)                                ORB16870
CALL RANGMO(R,FS,FW,DH)                                               ORB16880
RETURN
END

*****SUBROUTINE RANGMO(R,FS,FW,DH)
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE               ORB16890
* ANGULAR MOMENTUM                                                       ORB16900
* DOUBLE PRECISION FS,FW,DHW,DHS,DH,R                                 ORB16910
* DHW = R * FS                                                       ORB16920
* DHS = R * FW                                                       ORB16930
* DH = DSQRT((DHW**2) + (DHS**2))                                     ORB16940
* RETURN
* END

*****SUBROUTINE RSMAX(E,MM,R,A,FR,FS,DA,TA)
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE SEMI-MAJOR    ORB16950
* AXIS                                                               ORB16960
* DOUBLE PRECISION DA,FR,FS,E,MM,R,A,TA,ET                           ORB16970
* TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO                         ORB16980
* IF (E.GT. 0.9) THEN
*   ET = 0.9
* ELSE
*   ET = E
* ENDIF
*   DA = ((2.0D+00*E *DSIN(TA))/(MM *DSQRT(1.0D+00-(ET**2))))*FR +
* +   ((2.0D+00*A*DSQRT(1.0D+00-(E **2)))/(MM *R))*FS
* RETURN
* END

*****
```

```

* SUBROUTINE RECC(E,MM,R,A,FR,FS,TA,DE) ORB17300
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE ECCENTRICITY ORB17310
* DOUBLE PRECISION DE,FR,FS,E,MM,R,A,TA,ET ORB17320
* ORB17330
* ORB17340
* TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO ORB17350
* IF (E.LT.0.1) THEN ORB17360
*   ET = 0.1 ORB17370
* ELSE ORB17380
*   ET = E ORB17390
* ENDIF ORB17400
*   DE = ((DSQRT(1.0D+00 - (E **2))*SIN(TA))/(MM *A))*FR + ORB17410
*   + ((DSQRT(1.0D+00 - (E **2)))/(MM *ET*(A**2)))* ORB17420
*   + ((A**2)*(1.0D+00 - (E **2))/(R) - (R))*FS ORB17430
*   RETURN ORB17440
* END ORB17450
***** ORB17460
***** ORB17470
***** ORB17480
* SUBROUTINE RLANCE(MM,R,A,I,FW,AL,DLAN) ORB17490
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE LONGITUDE ORB17500
* OF THE ASCENDING NODE ORB17510
* ORB17520
* DOUBLE PRECISION DLAN,FW,E,MM,R,A,I,AL,ET,IT ORB17530
* ORB17540
* TRAP (E) AND (I) SO DENOMINATOR DOES NOT GOTO ZERO ORB17550
* IF (E.GT.0.9) THEN ORB17560
*   ET = 0.9 ORB17570
* ELSE ORB17580
*   ET = E ORB17590
* ENDIF ORB17600
* IF (I.LT.0.01745) THEN ORB17610
*   IT = 0.01745 ORB17620
* ELSE ORB17630
*   IT = I ORB17640
* ENDIF ORB17650
*   DLAN = (R*FW*DSIN(AL))/(MM *(A**2)*DSQRT(1.0D+00 - (ET**2))* ORB17660
*   + DSIN(IT)) ORB17670
*   RETURN ORB17680
* END ORB17690
* ORB17700
***** ORB17710
***** ORB17720
* SUBROUTINE RAP(E,MM,R,A,I,H,P,AL,TA,FR,FS,FW,DAP) ORB17730
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE ARGUMENT ORB17740
* OF PERIGEE ORB17750
* ORB17760
* DOUBLE PRECISION DAPR,DAPS,DAPW,DAP,FR,FS,FW,E,MM,R,I,H,P,AL,TA, ORB17770
* + ET,A,IT ORB17780
* ORB17790
* TRAP (E) AND (I) SO DENOMINATOR DOES NOT GOTO ZERO ORB17800
* IF (I.LT.0.01745) THEN ORB17810
*   IT = 0.01745 ORB17820
* ELSE ORB17830
*   IT = I ORB17840
* ENDIF ORB17850

```

```

IF (E.GT.0.9) THEN ORB17860
    ET = 0.9 ORB17870
ELSEIF (E.LT.0.1) THEN ORB17880
    ET = 0.1 ORB17890
ELSE ORB17900
    ET = E ORB17910
ENDIF ORB17920
DAPR = (-DSQRT(1.0+00 - (E **2))*DCOS(TA))/(MM *A*ET) * FR ORB17930
DAPS = (P/(ET*H))*(DSIN(TA))** ORB17940
+ (1.0D+00 + 1.0D+00/(1.0D+00 + ET*DCOS(TA))) *FS ORB17950
DAPW = (-R*(1.0D+00/DTAN(IT))*DSIN(AL))/ ORB17960
+ (MM *(A**2)*DSQRT(1.0D+00 - (ET**2)))**FW ORB17970
DAP = DAPR + DAPS + DAPW ORB17980
RETURN ORB17990
END ORB18000
*****
SUBROUTINE RINC(E,MM,R,A,FW,AL,DI) ORB18010
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE INCLINATION ORB18020
DOUBLE PRECISION DI,FW,E,MM,R,A,AL,ET ORB18030
* TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO ORB18040
IF (E.GT.0.9) THEN ORB18050
    ET = 0.9 ORB18060
ELSE ORB18070
    ET = E ORB18080
ENDIF ORB18090
DI = (R*FW*DCOS(AL))/(MM *(A**2)*DSQRT(1.0D+00 - (ET**2))) ORB18100
RETURN ORB18110
E.D ORB18120
*****
SUBROUTINE RMN(MM,A,DMM,DA,MU) ORB18130
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE MEAN MOTION ORB18140
DOUBLE PRECISION DMM,DA,MM,A,MU ORB18150
DMM = ((-3.0D+00*MU)/(2.0D+00*MM *(A**4)))* DA ORB18160
RETURN ORB18170
END ORB18180
*****
SUBROUTINE RMA(E,MM,R,A,TA,DMM,FR,FS,DMA,T) ORB18190
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE MEAN Anomaly ORB18200
DOUBLE PRECISION DMAA,DMAB,DMAC,DMAD,DMM,FR,FS,DMA,E,MM,R,A,TA, ORB18210
+ ET,T ORB18220
* TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO ORB18230
IF (E.GT.0.9) THEN ORB18240
    ET = 0.9 ORB18250
ELSEIF (E.LT.0.1) THEN ORB18260

```

```

        ET = 0.1          ORB18420
ELSE          ORB18430
    ET = E          ORB18440
ENDIF          ORB18450
DMM = (-1.0D+00/(MM *A))** ORB18460
+   (((2.0D+00*R)/A) - ((1 - (E **2))/ET)*DCOS(TA)) * FR - ORB18470
+   (1-(E **2))/(MM *A*ET)*(1+ R/(A*(1-(E**2))))*(SIN(TA)*FS) - ORB18480
+   (T * DMM)          ORB18490
RETURN          ORB18500
END            ORB18510
*****          ORB18520
* CALCULATE THE NEW ORBITAL ELEMENTS          ORB18530
*****          ORB18540
*****          ORB18550
*****          ORB18560
SUBROUTINE NSMA(A,DT,DA,NEWA)          ORB18570
* THIS SUBROUTINE CALCULATES THE NEW SEMI-MAJOR AXIS          ORB18580
DOUBLE PRECISION DA,DT,A,NEWA          ORB18590
NEWA = A + DA*DT          ORB18600
RETURN          ORB18610
END            ORB18620
*****          ORB18630
*****          ORB18640
*****          ORB18650
SUBROUTINE NECC(E,DT,DE,NEWE)          ORB18660
* THIS SUBROUTINE CALCULATES THE NEW ECCENTRICITY          ORB18670
DOUBLE PRECISION DE,DT,E,NEWE          ORB18680
NEWE = E + DE*DT          ORB18690
RETURN          ORB18700
END            ORB18710
*****          ORB18720
*****          ORB18730
*****          ORB18740
*****          ORB18750
*****          ORB18760
SUBROUTINE NINCL(I,DT,DI,NEWI)          ORB18770
* THIS SUBROUTINE CALCULATES THE NEW INCLINATION          ORB18780
DOUBLE PRECISION DI,DT,I,NEWI          ORB18790
NEWI = I + DI*DT          ORB18800
RETURN          ORB18810
END            ORB18820
*****          ORB18830
*****          ORB18840
*****          ORB18850
*****          ORB18860
*****          ORB18870
SUBROUTINE NASNOD(LAN,DT,DLAN,NEWLAN)          ORB18880
* THIS SUBROUTINE CALCULATES THE NEW LONGITUDE OF THE ASCENDING NODE          ORB18890
DOUBLE PRECISION DLAN,DT,LAN,NEWLAN          ORB18900
NEWLAN = LAN + DLAN*DT          ORB18910
RETURN          ORB18920
END            ORB18930
*****          ORB18940
*****          ORB18950
*****          ORB18960
*****          ORB18970

```

```

***** SUBROUTINE NARPER(AP,DT,DAP,NEWAP) ORB18980
* THIS SUBROUTINE CALCULATES THE NEW ARGUMENT OF PERIGEE ORB18990
DOUBLE PRECISION DAP,DT,AP,NEWAP ORB19000
NEWAP = AP + DAP*DT ORB19010
RETURN ORB19020
END ORB19030
***** SUBROUTINE NMNAN(MA,MM,DT,TF,DMA,PI) ORB19040
* THIS SUBROUTINE CALCULATES THE NEW MEAN Anomaly ORB19050
DOUBLE PRECISION DMM,FR,FS,DMA,DT,MA,E,R,A,TA,MM,TF,T,PI ORB19060
MA = MM*(TF) + DMA*DT ORB19070
IF (MA .GT. (2*PI)) THEN ORB19080
MA = MA - (2*PI) ORB19090
ENDIF ORB19100
RETURN ORB19110
END ORB19120
***** SUBROUTINE NMNMO(MM,DMM,DT,NEWMM) ORB19130
* THIS SUBROUTINE CALCULATE THE NEW MEAN MOTION ORB19140
DOUBLE PRECISION DMM,DT,MM,NEWMM ORB19150
NEWMM = MM + DMM*DT ORB19160
RETURN ORB19170
END ORB19180
***** SUBROUTINE NEA(MA,E,EA) ORB19190
* THIS SUBROUTINE CALCULATES THE NEW ECCENTRIC ANOMOLY BY USING ORB19200
NEWTONS METHOD OF ROOT FINDING ORB19210
DOUBLE PRECISION EAN,MAN,MA,E,EA,DIFF ORB19220
LET (EA) EQUAL (MA) FOR INITIAL GUESS AT ROOT ORB19230
EA = MA ORB19240
EAN = EA + (MA - EA + E*DSIN(EA))/(1.0D+00 - E*DCOS(EA)) ORB19250
MAN = EAN - E*SIN(EAN) ORB19260
CHECK DIFFERENCE (DIFF) ORB19270
DIFF = ABS(MA - MAN) ORB19280
EA = EAN ORB19290
CONTINUE TO INTERATE UNTIL DIFFERENCE IS NEGIGIBLE ORB19300
200 IF(DIFF.GT.0.000000001) THEN ORB19310
ORB19320
ORB19330
ORB19340
ORB19350
ORB19360
ORB19370
ORB19380
ORB19390
ORB19400
ORB19410
ORB19420
ORB19430
ORB19440
ORB19450
ORB19460
ORB19470
ORB19480
ORB19490
ORB19500
ORB19510
ORB19520
ORB19530

```

```

EAN = EA + (MA - EA + E*DSIN(EA))/(1.0D+00 - E*DCOS(EA))          ORB19540
MAN = EAN - E*DSIN(EAN)                                              ORB19550
EA = EAN                                                               CRB19560
DIFF = ABS(MA - MAN)                                                 ORB19570
GOTO 200                                                               ORB19580
ENDIF                                                               ORB19590
EA = EAN                                                               ORB19600
RETURN                                                               ORB19610
END                                                               ORB19620
ORB19630
ORB19640
ORB19650
ORB19660
ORB19670
ORB19680
ORB19690
ORB19700
ORB19710
ORB19720
ORB19730
ORB19740
ORB19750
ORB19760
ORB19770
ORB19780
ORB19790
ORB19800
ORB19810
ORB19820
ORB19830
ORB19840
ORB19850
ORB19860
ORB19870
ORB19880
ORB19890
ORB19900
ORB19910
ORB19920
ORB19930
ORB19940
ORB19950
ORB19960
ORB19970
ORB19980
ORB19990
ORB20000
ORB20010
ORB20020
ORB20030
ORB20040
ORB20050
ORB20060
ORB20070
ORB20080

***** SUBROUTINE NTA(EA,E,TA,PI)
* THIS SUBROUTINE CALCULATES THE NEW TRUE Anomaly
  DOUBLE PRECISION EA,E,TA,PI
    TA = DCOS((E - DCOS(EA))/(E*DCOS(EA) - 1.0D+00))
    IF (EA.GT.PI) THEN
      TA = (2*PI) - TA
    ENDIF
    RETURN
  END
***** SUBROUTINE NANGMO(H,DT,DH,NEWH)
* THIS SUBROUTINE CALCULATES THE NEW ANGULAR MOMENTUM
  DOUBLE PRECISION DH,DT,H,NEWH
    NEWH = H + DH*DT
    RETURN
  END
***** SUBROUTINE INTSUM(TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDMM,TDMA,
*                         TDLAN,TDH,TDAP)
* THIS SUBROUTINE INITIALIZES THE SUMS OF FORCES AND ELEMENT CHANGES
  TFEA = 0.0
  TFSU = 0.0
  TFMO = 0.0
  TFDRA = 0.0
  TDI = 0.0
  TDA = 0.0
  TDE = 0.0
  TDMM = 0.0
  TDMA = 0.0
  TDLAN = 0.0
  TDH = 0.0
  TDAP = 0.0
  RETURN
  END

```

```

***** *****
* CALCULATE THE NEW POSITION AND VELOCITY VECTORS
***** *****
* SUBROUTINE NPOS(RI,RJ,RK,R,LAN,AP,INC, TA,A,E)
* THIS SUBROUTINE CALCULATES THE NEW POSITION VECTOR
    DOUBLE PRECISION XW,YW,ZW,INC,RI,RJ,RK,R,LAN,AP,TA,A,E
* CALCULATE POSITION VECTOR IN 'PQW' SYSTEM
    R = (A*(1 - (E**2)))/(1 + E*DCOS(TA))
    XW = R*DCOS(TA)
    YW = R*DSIN(TA)
    ZW = 0
* TRANSFORM POSITION TO 'IJK' SYSTEM
    CALL PQWIJK(LAN,AP,INC,XW,YW,ZW,RI,RJ,RK)
    R = DSQRT((RI**2) + (RJ**2) + (RK**2))
    RETURN
    END
***** *****
* SUBROUTINE NVEL(E,P,TA,LAN,AP,INC,VI,VJ,VK,V,MU)
* THIS SUBROUTINE CALCULATES THE NEW VELOCITY VECTOR
    DOUBLE PRECISION INC,VP,VQ,VW,MU,E,P,TA,LAN,AP,VI,VJ,VK,V
* CALCULATE VELOCITY IN 'PQW' SYSTEM
    VP = DSQRT(MU/P)*(-DSIN(TA))
    VQ = DSQRT(MU/P)*(E + DCOS(TA))
    VW = 0.0D+00
* TRANSFORM VELOCITY INTO 'IJK' SYSTEM
    CALL PQWIJK(LAN,AP,INC,VP,VQ,VW,VI,VJ,VK)
    V = DSQRT((VI**2) + (VJ**2) + (VK**2))
    RETURN
    END
***** *****
* VELOCITY CHANGE
***** *****
SUBROUTINE CHGVEL(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,
+ VI,VJ,VK,V,MU,PI,H,A,E,N,TA,P,MM,MA,EA,
+ TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,ZARAY,AINRAY,APRAY,TIMRAY,
+ TT,EI,EJ,EK,LP,HI,HJ,IOPT1,TFEA,TFSU,TFMO,TFDRA,
+ TDI,TDA,TDE,TDMM,TDMA,TDLAN,TDH,TDAP)
* THIS SUBROUTINE CALCULATE VELOCITY CHANGES
* THE FOLLOWING SUBROUTINES ARE CALLED:
* TACHG = RETURNS TRUE ANOMALY FOR VELOCITY CHANGE LOCATION (CHTA) ORB20610
* AND AN INDICATOR OF LOCATION (ITA) ORB20620
* CALCEL = CALCULATE Orbital ELEMENTS ORB20630
* UNPREZ = CALCULATE UNPERTURBED ORBIT ORB20640

```

*	NPOS = CALCULATE NEW POSITION	ORB20650
*	NVEL = CALCULATE NEW VELOCITY	ORB20660
*	STORE = STORE POSITION AND ELEMENTS IN ARRAYS	ORB20670
*	ENERGY = ENERGY OF SATELLITE	ORB20680
*	ECC = ECCENTRICITY	ORB20690
*	SXANIS = SEMI-MAJOR AXIS	ORB20700
	DOUBLE PRECISION T,DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, + MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,TT, + NEWVI,NEWVJ,NEWVK,NEWV,VMAX,CNTA,EI,EJ,EX,LP,HI,HJ,VCIR, + DI,DE,DA,DMM,DMA,DLAN,DH,DAP,NEWEI,NEWEJ,NEWEK,NEWE,NEWENR, + NEWA,NEWRP,RE	ORB20710 ORB20720 ORB20730 ORB20740 ORB20750 ORB20760 ORB20770
	DIMENSION RARAY(500),TARAY(500),RIRAY(500),RJRAY(500), + RKRAY(500),AINRAY(500),APRAY(500),TIMRAY(500)	ORB20780 ORB20790 ORB20800
	CHARACTER*1,YORN,PYORN	ORB20810 ORB20820 ORB20830
	RE = 6.3782D+03	ORB20840
*	PROMPT THE USER FOR THE "VELOCITY Change LOCATION CALL TACNG(PI,CHTA,ITA)	ORB20850 ORB20860 ORB20870
*	SET TIME COUNTER TO ONE TIME STEP T = DT	ORB20880 ORB20890 ORB20900
*	ROTATE TO THE VELOCITY CHANGE LOCATION THIS IS IDENTICAL TO THE Unperturbed ORBIT WITH THE EXCEPTION THAT A COMPLETE ORBIT IS NOT CALCULATED PRINT*, 'ROTATE TO VELOCITY CHANGE LOCATION' IF ((ITA.EQ.2) .OR. (ITA.EQ.3)) THEN	ORB20910 ORB20920 ORB20930 ORB20940 ORB20950
	PRINT*, 'BEFORE TA =', TA IF (TA .GT. 6.21) THEN TA = TA - (2*PI)	ORB20960 ORB20970 ORB20980
	ENDIF	ORB20990
250	IF((T.LE.PER).AND.(TA.LT.CHTA)) THEN PRINT*, 'TA =', TA NUM = NUM + 1 TT = TT + DT CALL NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER) CALL NPOS(RI,RJ,RK,R,LAN,AP,I,TA,A,E) CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU) CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY, + TARAY,NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB21000 ORB21010 ORB21020 ORB21030 ORB21040 ORB21050 ORB21060 ORB21070 ORB21080
	T = T + DT GOTO 250	ORB21090 ORB21100
	ENDIF	ORB21110
	IF (TF .GE. PER) THEN TF = TF - PER	ORB21120 ORB21130
	ENDIF	ORB21140
	ENDIF	ORB21150
*	PRINT ESCAPE VELOCITY AND CIRCULAR VELOCITY FOR Reference CALL EXCNS('CLRSCRN') PRINT*, 'AFTER TA =', TA PRINT*, 'THIS SHOULD BE THE DESIRED RADIUS RP OR RA'	ORB21160 ORB21170 ORB21180 ORB21190 ORB21200

```

260 PRINT*, 'RADIUS = ', R
PRINT*, 'VELOCITY = ', V
VMAX = DSQRT(2.0*(MU / R))
PRINT*, 'MAX VELOCITY AT THIS RADIUS IS: ', VMAX
VCIR = DSQRT(MU/R)
PRINT*, 'CIRCULAR VELOCITY AT THIS RADIUS IS : ', VCIR
                                         ORB21210
                                         ORB21220
                                         ORB21230
                                         ORB21240
                                         ORB21250
                                         ORB21260
                                         ORB21270
                                         ORB21280
                                         ORB21290
                                         ORB21300
                                         ORB21310
                                         ORB21320
                                         ORB21330
                                         ORB21340
                                         ORB21350
                                         ORB21360
                                         ORB21370
                                         ORB21380
                                         ORB21390
                                         ORB21400
                                         ORB21410
                                         ORB21420
                                         ORB21430
                                         ORB21440
                                         ORB21450
                                         ORB21460
                                         ORB21470
                                         ORB21480
                                         ORB21490
                                         ORB21500
                                         ORB21510
                                         ORB21520
                                         ORB21530
                                         ORB21540
                                         ORB21550
                                         ORB21560
                                         ORB21570
                                         ORB21580
                                         ORB21590
                                         ORB21600
                                         ORB21610
                                         ORB21620
                                         ORB21630
                                         ORB21640
                                         ORB21650
                                         ORB21660
                                         ORB21670
                                         ORB21680
                                         ORB21690
                                         ORB21700
                                         ORB21710
                                         ORB21720
                                         ORB21730
                                         ORB21740
                                         ORB21750
                                         ORB21760
*   PROMPT USER TO CHANGE VELOCITY IN ORBITAL PLANE
PRINT*, 'DO YOU WANT TO CHANGE THE VELOCITY IN THE ORBITAL PLANE?'
PRINT*, 'ENTER "Y" OR "N" :'
READ*, PYORN
PRINT*, PYORN
IF (PYORN .EQ. 'Y') THEN
    PRINT*, 'GIVE THE TOTAL CHANGE IN VELOCITY, I. E. 5.0 KM.'
    PRINT*, 'THE PROGRAM WILL FIGURE OUT THE FINAL VELOCITY VECTOR'
    PRINT*, ' ENTER VELOCITY CHANGE:'
    READ*, CHGV
    PRINT*, CHGV
                                         ORB21210
                                         ORB21220
                                         ORB21230
                                         ORB21240
                                         ORB21250
                                         ORB21260
                                         ORB21270
                                         ORB21280
                                         ORB21290
                                         ORB21300
                                         ORB21310
                                         ORB21320
                                         ORB21330
                                         ORB21340
                                         ORB21350
                                         ORB21360
                                         ORB21370
                                         ORB21380
                                         ORB21390
                                         ORB21400
                                         ORB21410
                                         ORB21420
                                         ORB21430
                                         ORB21440
                                         ORB21450
                                         ORB21460
                                         ORB21470
                                         ORB21480
                                         ORB21490
                                         ORB21500
                                         ORB21510
                                         ORB21520
                                         ORB21530
                                         ORB21540
                                         ORB21550
                                         ORB21560
                                         ORB21570
                                         ORB21580
                                         ORB21590
                                         ORB21600
                                         ORB21610
                                         ORB21620
                                         ORB21630
                                         ORB21640
                                         ORB21650
                                         ORB21660
                                         ORB21670
                                         ORB21680
                                         ORB21690
                                         ORB21700
                                         ORB21710
                                         ORB21720
                                         ORB21730
                                         ORB21740
                                         ORB21750
                                         ORB21760
*   CALCULATE NEW VELOCITY FOR CHANGE IN THE ORBITAL PLANE
NEWVI = VI + (CHGV * VI / V)
NEWVJ = VJ + (CHGV * VJ / V)
NEWVK = VK + (CHGV * VK / V)
                                         ORB21210
                                         ORB21220
                                         ORB21230
                                         ORB21240
                                         ORB21250
                                         ORB21260
                                         ORB21270
                                         ORB21280
                                         ORB21290
                                         ORB21300
                                         ORB21310
                                         ORB21320
                                         ORB21330
                                         ORB21340
                                         ORB21350
                                         ORB21360
                                         ORB21370
                                         ORB21380
                                         ORB21390
                                         ORB21400
                                         ORB21410
                                         ORB21420
                                         ORB21430
                                         ORB21440
                                         ORB21450
                                         ORB21460
                                         ORB21470
                                         ORB21480
                                         ORB21490
                                         ORB21500
                                         ORB21510
                                         ORB21520
                                         ORB21530
                                         ORB21540
                                         ORB21550
                                         ORB21560
                                         ORB21570
                                         ORB21580
                                         ORB21590
                                         ORB21600
                                         ORB21610
                                         ORB21620
                                         ORB21630
                                         ORB21640
                                         ORB21650
                                         ORB21660
                                         ORB21670
                                         ORB21680
                                         ORB21690
                                         ORB21700
                                         ORB21710
                                         ORB21720
                                         ORB21730
                                         ORB21740
                                         ORB21750
                                         ORB21760
*   Velocity CHANGE OUT OF ORBITAL PLANE
ELSEIF (PYORN .EQ. 'N') THEN
    PRINT*, ' ENTER THE NEW VELOCITY VECTOR:'
    PRINT*, ' ENTER THE NEW VI'
    READ*, NEWVI
    PRINT*, NEWVI
    PRINT*, ' ENTER THE NEW VJ'
    READ*, NEWVJ
    PRINT*, NEWVJ
    PRINT*, ' ENTER THE NEW VK'
    READ*, NEWVK
    PRINT*, NEWVK
    NUM = 1
    ITA = 3
ELSE
    CALL EXCMS('GLRSCRN')
    GOTO 260
ENDIF
*   PRINT NEW VELOCITY FOR USER TO CHECK
NEWV = DSQRT((NEWVI**2) + (NEWVJ**2) + (NEWVK**2))
PRINT*, 'NEW VI = ', NEWVI
PRINT*, 'NEW VJ = ', NEWVJ
PRINT*, 'NEW VK = ', NEWVK
PRINT*, 'NEW V = ', NEWV
PRINT*, 'ARE THESE VALUES THE ONES YOU WANT?'
PRINT*, 'ENTER "Y" OR "N" :'
READ*, YORN
PRINT*, YORN
IF (YORN .EQ. 'N') THEN
    CALL EXCMS('CLRSCRN')
    GOTO 260
                                         ORB21210
                                         ORB21220
                                         ORB21230
                                         ORB21240
                                         ORB21250
                                         ORB21260
                                         ORB21270
                                         ORB21280
                                         ORB21290
                                         ORB21300
                                         ORB21310
                                         ORB21320
                                         ORB21330
                                         ORB21340
                                         ORB21350
                                         ORB21360
                                         ORB21370
                                         ORB21380
                                         ORB21390
                                         ORB21400
                                         ORB21410
                                         ORB21420
                                         ORB21430
                                         ORB21440
                                         ORB21450
                                         ORB21460
                                         ORB21470
                                         ORB21480
                                         ORB21490
                                         ORB21500
                                         ORB21510
                                         ORB21520
                                         ORB21530
                                         ORB21540
                                         ORB21550
                                         ORB21560
                                         ORB21570
                                         ORB21580
                                         ORB21590
                                         ORB21600
                                         ORB21610
                                         ORB21620
                                         ORB21630
                                         ORB21640
                                         ORB21650
                                         ORB21660
                                         ORB21670
                                         ORB21680
                                         ORB21690
                                         ORB21700
                                         ORB21710
                                         ORB21720
                                         ORB21730
                                         ORB21740
                                         ORB21750
                                         ORB21760

```

```

        ENDIF ORB21770
*     CHECK FOR VALID VELOCITY ORB21780
    IF ( NEWV .GT. VMAX) THEN ORB21790
        PRINT*, 'YOUR VELOCITY IS TO GREAT !!' ORB21800
        GOTO 260 ORB21810
    ENDIF ORB21820
*     Calculate PERIGEE RADIUS TO SEE IF SATELLITE WILL IMPACT EARTH ORB21830
    CALL ENERGY(NEWV,R,MU,NEWENR) ORB21840
    CALL ECC(RI,RJ,RK,R,NEWVI,NEWVJ,NEWVK,NEWV,NEWEI,NEWEJ,NEWEK, + ORB21850
    +      NEWE,MU) ORB21860
    CALL SMAXIS(MU,NEWENR,NEWA) ORB21870
    NEWRP = NEWA*(1.0 - NEWE) ORB21880
    IF (NEWRP .LE. RE) THEN ORB21890
        PRINT*, 'YOUR VELOCITY AT THIS POINT IS TO SMALL!!!' ORB21900
        PRINT*, 'THE SATELLITE WILL IMPACT THE EARTH!!' ORB21910
        PRINT*, 'THE SATELLITES RADIUS OF PERIGEE WOULD BE ',NEWRP ORB21920
        PRINT*, 'A NEW VELOCITY WILL HAVE TO BE ENTERED!!' ORB21930
        GOTO 260 ORB21940
    ENDIF ORB21950
*     ACCEPT NEW VELOCITY ORB21960
    VI = NEWVI ORB21970
    VJ = NEWVJ ORB21980
    VK = NEWVK ORB21990
    V = NEWV ORB22000
*     CALCULATE NEW ELEMENT WITH NEW VELOCITY AND SET TIME STEP ORB22010
    CALL CALCEL( RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,LP,TA, + ORB22020
    +      PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ) ORB22030
    DT = PER/50.0 ORB22040
    T = DT ORB22050
*     THE FOUR Different CASES OF VELOCITY CHANGES FOLLOWS: ORB22060
*     VELOCITY CHANGE AT PERIGEE, AND NEWV > V Circular ORB22070
    IF((ITA.EQ.1).AND.(NEWV.GT.VCIR))THEN ORB22080
        CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, + ORB22090
        +      MU,PI,H,A,E,N,TA,P,MM, ORB22100
        +      MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY, ORB22110
        +      TARAY,AINRAY,APRAY,TIMRAY,TT) ORB22120
*     Change VELOCITY AT PERIGEE, AND NEWV <= V CIRCULAR ORB22130
*     APUGEE AND PERIGEE SWAP ORB22140
    ELSEIF ((ITA.EQ.1).AND.(NEWV.LE.VCIR))THEN ORB22150
        CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY, + ORB22160
        +      NUM,I,AP,AINRAY,APRAY,TT,TIMRAY) ORB22170
        T = PER/2 ORB22180
*     STEP SATELLITE TO NEW PERIGEE, ONLY A HALF ORBIT ORB22190
    CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, + ORB22200
        +      MU,PI,H,A,E,N,TA,P,MM, ORB22210
        +      ORB22220
        +      ORB22230
        +      ORB22240
        +      ORB22250
        +      ORB22260
        +      ORB22270
        +      ORB22280
        +      ORB22290
        +      ORB22300
        +      ORB22310
        +      ORB22320

```

```

+                               MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,
+                               TARAY,AINRAY,APRAY,TIMRAY,TT)
+
*      RESET TIME COUNTER TO ONE TIME STEP
*      T = DT
+
*      CALCULATE COMPLETE NEXT ORBIT
*      CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,
+                               MU,PI,H,A,E,N,TA,P,MM,
+                               MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,
+                               TARAY,AINRAY,APRAY,TIMRAY,TT)
+
*      CHANGE VELOCITY AT APOGEE, AND NEW V < V CIRCULAR
*      ELSEIF ((ITA.EQ.2) .AND.(NEWV .LT. VCIR)) THEN
+
*          T = PER/2
+
*          FINISH ORBIT
*          CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,
+                               MU,PI,H,A,E,N,TA,P,MM,
+                               MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,
+                               TARAY,AINRAY,APRAY,TIMRAY,TT)
+
*          CHANGE VELOCITY AT Apogee, AND NEWV >= V CIRCULAR
*          OR AT ANY OTHER TRUE Anomaly
*          ELSEIF (((ITA.EQ.2).AND.(NEWV.GE.VCIR)) .OR. (ITA.EQ.3)) THEN
+
*              IF (TA .GT. 6.21) THEN
*                  TA = TA - (2*PI)
*              ENDIF
+
*              CLEAR PREVIOUS ORBITS AND STEP SATELLITE TO NEW PERIGEE
*              T = TF
*              NUM = 1
*              CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,
+                               NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)
*              CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,
+                               MU,PI,H,A,E,N,TA,P,MM,
+                               MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,
+                               TARAY,AINRAY,APRAY,TIMRAY,TT)
*
*              IF (TF .GE. PER) THEN
*                  TF = TF - PER
*              ENDIF
+
*              CALCULATE COMPLETE NEXT ORBIT
*              T = DT
*              CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,
+                               MU,PI,H,A,E,N,TA,P,MM,
+                               MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,
+                               TARAY,AINRAY,APRAY,TIMRAY,TT)
*
*              ENDIF
*              RETURN
*          END

```

```

* SUBROUTINE TACNG(PI, CHTA, ITA) ORB22890
* THIS SUBROUTINE ASKS THE USER FOR VELOCITY CHANGE LOCATION ORB22900
* DOUBLE PRECISION CHTA, PI ORB22910
* ORB22920
* CALL EXCMS('CLRSCRN') ORB22930
* PRINT*, 'WHERE DO YOU WANT TO CHANGE THE VELOCITY?' ORB22940
* PRINT*, ' 1. AT CURRENT PERIGEE' ORB22950
* PRINT*, ' 2. AT CURRENT Apogee' ORB22960
* PRINT*, ' 3. AT A SPECIFIC TRUE Anomaly' ORB22970
* PRINT*, 'ENTER "1", "2" OR "3"' ORB22980
* READ*, ITA ORB22990
* PRINT*, ITA ORB23000
* ORB23010
* ORB23020
* SET TRUE ANOMALY CHANGE LOCATION (CHTA) TO DESIRED LOCATION ORB23030
* IF (ITA .EQ. 1) THEN ORB23040
*   CHTA = 0.0 ORB23050
* ENDIF ORB23060
* IF (ITA .EQ. 2) THEN ORB23070
*   CHTA = PI ORB23080
* ENDIF ORB23090
* IF (ITA .EQ. 3) THEN ORB23100
*   PRINT*, 'AT WHAT TRUE ANOMALY DO YOU WANT TO CHANGE THE' ORB23110
*   PRINT*, 'VELOCITY?' ORB23120
*   PRINT*, 'ENTER TRUE ANOMALY IN DEGREES' ORB23130
*   READ*, CHTA ORB23140
*   PRINT*, CHTA ORB23150
*   CHTA = CHTA * PI / 180 ORB23160
* ENDIF ORB23170
* RETURN ORB23180
* END ORB23190
* ORB23200
* ORB23210
* * OUTPUT PLOTS ORB23220
* * ORB23230
* * ORB23240
* SUBROUTINE PLOTS(RIRAY, RJRAY, RKRAY, RARAY, TARAY, NUM, PI, INC, LP, A, ORB23250
* + E, TF, AINRAY, APRAY, TIMRAY, TFEA, TFSU, TFMO, TFDRA, ORB23260
* + PER, TDI, TDA, TDE, TDMM, TDMA, TDLAN, TDH, TDAP, ORB23270
* + MM, MA, LAN, H, AP, R, V) ORB23280
* ORB23290
* THIS SUBROUTINE ASKS THE USER FOR THE TYPE OF OUTPUT THAT IS ORB23300
* DESIRED PERIFOCAL, GROUND TRACK OR TO SKIP THE PLOT. ORB23310
* ORB23320
* THE FOLLOWING SUBROUTINES ARE CALLED: ORB23330
* PERIF = PLOT PERIFOCAL ORBIT ORB23340
* GRTRK = PLOT GROUND TRACK ORB23350
* DATE = DISPLAYS DATA ON PLOT ORB23360
* TEC618 = SET Disspla TO TEC 618 OUTPUT ORB23370
* ENDPL = END THIS DISSPLA PLOT ORB23380
* REFER TO DISSPLA USER'S MANUAL FOR EXPLANATION OF DISSPLA ORB23390
* SUBROUTINES ORB23400
* ORB23410
* DOUBLE PRECISION PI, A, E, INC, LP, TF, PER, MM, MA, LAN, H, AP, R, V ORB23420
* ORB23430
* DIMENSION RIRAY(500), RJRAY(500), RKRAY(500), RARAY(500), TARAY(500), ORB23440

```

```

+     AINRAY(500),APRAY(500),TIMRAY(500)          ORB23450
CHARACTER#1,YORN                                ORB23460
CALL EXCN8('CLRSRGN')                            ORB23470
ORB23480
ORB23490
ORB23500
ORB23510
ORB23520
ORB23530
ORB23540
ORB23550
ORB23560
ORB23570
ORB23580
ORB23590
ORB23600
ORB23610
ORB23620
ORB23630
ORB23640
ORB23650
ORB23660
ORB23670
ORB23680
ORB23690
ORB23700
ORB23710
ORB23720
ORB23730
ORB23740
ORB23750
ORB23760
ORB23770
ORB23780
ORB23790
ORB23800
ORB23810
ORB23820
ORB23830
ORB23840
ORB23850
ORB23860
ORB23870
ORB23880
ORB23890
ORB23900
ORB23910
ORB23920
ORB23930
ORB23940
ORB23950
ORB23960
ORB23970
ORB23980
ORB23990
ORB24000

* CALCULATE SINGLE PRECISION VARIABLES
  SPI = SNGL(PI)
  SA = SNGL(A)
  SE = SNGL(E)
  SINC = SNGL(INC)
  SLP = SNGL(LP)
  STF = SNGL(TF)
  SPER = SNGL(PER)
  SMM = SNGL(MM)
  SMA = SNGL(MA)
  SLAN = SNGL(LAN)
  SH = SNGL(H)
  SAP = SNGL(AP)
  SV = SNGL(V)
  SR = SNGL(R)

* PROMPT USER FOR DISPLAY TYPE
340 PRINT*, 'WHAT TYPE OF Display IS DESIRED:'
  PRINT*, '      1. PERIFOCAL'
  PRINT*, '      2. GROUND TRACK'
  PRINT*, '      3. SKIP PLOT'
  PRINT*, 'ENTER 1,2,3,4:'
  READ*, INPUT
  PRINT350, INPUT
350 FORMAT(I4)

  CALL TEK618

* CALL APPROPRIATE PLOT
  IF (INPUT .EQ. 1) THEN
    CALL PERIF(RARAY,TARAY,NUM,SPI,SINC,SLP,SA,SE)
  ELSEIF (INPUT .EQ. 2) THEN
    CALL GRTRK(AINRAY,APRAY,TARAY,STF,NUM,TIMRAY)
  ELSEIF (INPUT .EQ. 3) THEN
    GOFO 360
  ELSE
    PRINT*, 'INVALID ENTRY!'
    GOTO 340
  ENDIF

* DISPLAY DATA
  CALL DATA(SINC,SA,SE,TEFA,TFSU,TFMO,TFDRA,SPER,SPI,TDI,TDA,TDE,
+           TDMM,TDMA,TDLAN,TDH,TDAP,SMM,SMA,SLAN,SH,SAP,SV,SR)
  CALL ENDPL(0)

* PROMPT USER IF ANOTHER DISPLAY TYPE IS DESIRED
  PRINT*, 'WOULD YOU LIKE ANOTHER PLOT USING THE SAME ORBITAL'
  PRINT*, 'PARAMETERS AND DATA: '
  PRINT*, 'ENTER "Y" OR "N" : '
  READ*, YORN
  PRINT*, YORN

```

```

IF (YORN .EQ. 'Y') THEN          ORB24010
    GOTO 340                     ORB24020
ENDIF                           ORB24030
360  RETURN                      ORB24040
END                            ORB24050
                                ORB24060
*****                         ORB24070
                                ORB24080
SUBROUTINE PERIF(RARAY,TARAY,NUM,PI,INC,LP,A,E)   ORB24090
* THIS SUBROUTINE PLOTS OUT THE RESULTS OF THE PROGRAM USING THE   ORB24100
* DISPLAY FEATURE ON THE MAIN FRAME.                               ORB24110
* REFER TO DISSPLA USERS GUIDE FOR EXPLANATION OF DISSPLA      ORB24120
* SUBROUTINES.                                                 ORB24130
REAL INC,LP                  ORB24140
DIMENSION TARAY(500),RARAY(500),RIRAY(500),RJRAY(500)  ORB24150
I = 1                         ORB24160
                                ORB24170
* SET SCALE OF AXIS          ORB24180
RSTEP = (A^(1+E)) / 3        ORB24190
CALL TEN618                  ORB24200
CALL RESET(3HALL)            ORB24210
CALL SCMPIX                 ORB24220
CALL PHYSOR(1.25,4.)         ORB24230
CALL AREA2D(6.,6.)           ORB24240
CALL MESSAG('PERIFOCAL COORDINATE SYSTEM$',100,1.0,6.5)  ORB24250
CALL XNAME('XW',2)            ORB24260
CALL YNAME('YW',2)            ORB24270
CALL XANANG(90.0)             ORB24280
CALL YANANG(0.0)              ORB24290
CALL INTAXS                  ORB24300
CALL POLAR(1.,RSTEP,3.,3.)    ORB24310
CALL POLY3                   ORB24320
CALL NOCHEN                  ORB24330
CALL CURVE(TARAY,RARAY,NUM,1) ORB24340
CALL COMPLX                  ORB24350
CALL HEIGHT(.2)               ORB24360
CALL RESET('COMPLEX')        ORB24370
CALL RESET('HEIGHT')          ORB24380
CALL ENDGR(0)                 ORB24390
                                ORB24400
                                ORB24410
                                ORB24420
* Display EARTH PLOT          ORB24430
CALL EARTH1(A,E,INC,PI,RSTEP)  ORB24440
RETURN                        ORB24450
END                           ORB24460
                                ORB24470
*****                         ORB24480
                                ORB24490
SUBROUTINE EARTH1(A,E,INC,PI,RSTEP)    ORB24500
* THIS SUBROUTINE PLOTS A VIEW OF THE WORLD, LOOKING DOWN THE 'Z'   ORB24510
* AXIS, PLACED ON THE ORIGIN. THE Latitude IS FIXED, BUT THE       ORB24520
* LONGITUDE VARIES WITH THE INCLINATION.                          ORB24530
* REFER TO DISSPLA USER'S MANUAL FOR EXPLANATION OF DISSPLA     ORB24540
* SUBROUTINES.                                                 ORB24550
                                ORB24560

```

REAL INC	ORB24570
COMMON IWORK(3800)	ORB24580
DATA IWDIM/3800/	ORB24590
RE = 6378.145	ORB24600
* SCALE THE EARTH PLOT AND CENTER ON THE ORIGIN	ORB24610
SCFAC = RE/RSTEP	ORB24620
SCFAC2 = SCFAC * 2.0	ORB24630
XPHS = 1.25 + 3.0 - SCFAC	ORB24640
YPHS = 4.0 + 3.0 - SCFAC	ORB24650
YPOLE = 90 - (INC * 180 / PI)	ORB24660
IF(YPOLE .GT. 90) THEN	ORB24670
YPOLE = YPOLE - 90	ORB24680
ENDIF	ORB24690
YORIG = YPOLE - 90	ORB24700
YMAX = YPOLE + 90	ORB24710
CALL RESET(3HALL)	ORB24720
CALL PHYSOR(XPHS,YPHS)	ORB24730
CALL PROJCT('LAMBERT EQ/AREA')	ORB24740
CALL MAPOLE(0.0,YPOLE)	ORB24750
CALL AREA2D(SCFAC2,SCFAC2)	ORB24760
CALL THKFRM(0.02)	ORB24770
CALL GRAF(-90.,30.,90.,YORIG,30.,YMAX)	ORB24780
CALL FRAME	ORB24790
CALL MAPFIL('MAPDTA')	ORB24800
CALL LBLANK('LAND',IWDIM)	ORB24810
CALL GRID(1,1)	ORB24820
CALL LBLANK('WATER',IWDIM)	ORB24830
CALL DASH	ORB24840
CALL GRID(1,1)	ORB24850
CALL RESET('DASH')	ORB24860
CALL ENDGR(0)	ORB24870
RETUR.	ORB24880
END	ORB24890
*****	ORB24900
SUBROUTINE GRTRK(AINRAY,APRAY,TARAY,TF,NUM,TIMRAY)	ORB24910
DIMENSION AINRAY(500),APRAY(500),TARAY(500),	ORB24920
+ ELARAY(500),ELORAY(500),TLONG(500),TLAT(500),TIMRAY(500)	ORB24930
RE = 6.3782E+03	ORB24940
EROT = 7.292115856E-05	ORB24950
STF = (TF)	ORB24960
I = 1	ORB24970
* LOAD ARRAYS WITH LATITUDE AND LONGITUDE	ORB24980
410 IF (I .LE. NUM) THEN	ORB24990
X = RE*COS(APRAY(I))*COS(TARAY(I))-RE*SIN(APRAY(I))*	ORB25000
+ SIN(TARAY(I))	ORB25010
Y = RE*COS(AINRAY(I))*SIN(APRAY(I))*COS(TARAY(I)) +	ORB25020
+ RE*COS(AINRAY(I))*COS(APRAY(I))*SIN(TARAY(I))	ORB25030
Z = RE*SIN(AINRAY(I))*SIN(APRAY(I))*COS(TARAY(I)) +	ORB25040
+ RE*SIN(AINRAY(I))*COS(APRAY(I))*SIN(TARAY(I))	ORB25050

```

*      CALCULATE LATITUDE          ORB25130
ELARAY(I) = (ASIN(Z/RE)) * (180/3.14159)          ORB25140
*      TRAP 'X' AND 'Y' FOR ARCTAN IN CALCULATING LONGITUDE    ORB25150
IF((Y .LE. 10) .AND. (Y .GE. 0.0)) THEN          ORB25160
  Y = 10.
ELSEIF ((Y .GE. -10).AND.(Y .LE. 0.0)) THEN      ORB25170
  Y = -10.
ENDIF                                              ORB25180
IF((X .LE. 10) .AND. (X .GE. 0.0)) THEN          ORB25190
  X = 10.
ELSEIF ((X .GE. -10).AND.(X .LE. 0.0)) THEN      ORB25200
  X = -10.
ENDIF                                              ORB25210
*      CALCULATE LONGITUDE        ORB25220
ELORAY(I) = (ATAN2(Y,X) - (EROT*TIMRAY(I))) * (180/3.14159)  ORB25230
*      MODIFY LONGITUDES TO ( -180 TO 180)    ORB25240
420   IF (ELORAY(I) .LT. -180) THEN          ORB25250
      ELORAY(I) = ELORAY(I) + 360          ORB25260
      GOTO 420                          ORB25270
ENDIF                                              ORB25280
I = I + 1                                         ORB25290
GOTO 410                                         ORB25300
ENDIF                                              ORB25310
*      SET DISSPLA                ORB25320
CALL TEK61S                         ORB25330
CALL RESET(3HALL)                     ORB25340
CALL YAXANG (0.)                      ORB25350
CALL PHYSOR(1.0,6.0)                  ORB25360
CALL XNAME(' ',1)                    ORB25370
CALL YNAME(' ',1)                    ORB25380
CALL AREA2D(7.5,3.75)                ORB25390
CALL HEADIN ('GROUND TRACKS',100,1.5,1)  ORB25400
CALL SCMPLN                           ORB25410
CALL MAPGR(-180.,90.,180.,-90.,30.,90.)  ORB25420
CALL GRID (1,1)                      ORB25430
CALL MAPFIL ('MAPDTA')               ORB25440
I = 1                                     ORB25450
*      IGNORE Boundary POINTS     ORB25460
430   IF ((ELORAY(I) .LT. -175) .OR.          ORB25470
+      (ELORAY(I) .GT. 175) .OR.          ORB25480
+      (ELARAY(I) .LT. -85) .OR.          ORB25490
+      (ELARAY(I) .GT. 85)) THEN          ORB25500
  I = I + 1                            ORB25510
  GOTO 430                          ORB25520
ENDIF                                              ORB25530
ITEMP = 1                                         ORB25540
*      LOAD FIRST POINT OF NEW PLOT SEGMENT    ORB25550
IF (I .LE. NUM) THEN                         ORB25560

```

```

TLONG(ITEMP) = ELORAY(I)          ORB25690
TLAT(ITEMP) = ELARAY(I)          ORB25700
I = I + 1                         ORB25710
* IF ( I .GE. NUM) THEN          ORB25720
*   CALL POLY3                   ORB25730
*   CALL CURVE(TLONG,TLAT,ITEMP,1) ORB25740
* ENDIF                         ORB25750
ENDIF                           ORB25760
                                ORB25770
* LOAD SECOND POINT IN LINE SEGMENT ORB25780
IF ( I .LE. NUM) THEN           ORB25790
  ITEMP = ITEMP + 1             ORB25800
  TLONG(ITEMP) = ELORAY(I)      ORB25810
  TLAT(ITEMP) = ELARAY(I)      ORB25820
  I = I + 1                    ORB25830
  IF ( I .GE. NUM) THEN        ORB25840
    CALL POLY3                  ORB25850
    CALL NOCHEK                ORB25860
    CALL CURVE(TLONG,TLAT,ITEMP,1) ORB25870
  ENDIF                         ORB25880
ENDIF                           ORB25890
                                ORB25900
* LOOP UNTIL SEGMENT REACHES EDGE OR NO MORE POINTS ORB25910
440 IF ( I .LE. NUM) THEN       ORB25920
                                ORB25930
* BOTH LAT AND LONG INCREASING ORB25940
  IF((ELORAY(I - 2) .LE. ELORAY(I - 1)) .AND. ORB25950
+   (ELARAY(I - 2) .LE. ELARAY(I - 1))) THEN ORB25960
    IF((ELORAY(I) .LT. -170) .OR. ORB25970
+     (ELARAY(I) .LT. -80)) THEN ORB25980
      CALL POLY3                  ORB25990
      CALL NOCHEK                ORB26000
      CALL CURVE(TLONG,TLAT,ITEMP,1) ORB26010
      GOTO 430                   ORB26020
    ELSE                         ORB26030
      ITI = ITEMP + 1            ORB26040
      TLONG(ITEMP) = ELORAY(I)  ORB26050
      TLAT(ITEMP) = ELARAY(I)  ORB26060
    ENDIF                         ORB26070
                                ORB26080
* BOTH LAT AND LONG DECREASING ORB26090
ELSEIF((ELORAY(I - 2) .GT. ELORAY(I - 1)) .AND. ORB26100
+   (ELARAY(I - 2) .GT. ELARAY(I - 1))) THEN ORB26110
    IF((ELORAY(I) .GT. 170) .OR. ORB26120
+     (ELARAY(I) .GT. 80)) THEN ORB26130
      CALL POLY3                  ORB26140
      CALL NOCHEK                ORB26150
      CALL CURVE(TLONG,TLAT,ITEMP,1) ORB26160
      GOTO 430                   ORB26170
    ELSE                         ORB26180
      ITEMP = ITEMP + 1          ORB26190
      TLONG(ITEMP) = ELORAY(I)  ORB26200
      TLAT(ITEMP) = ELARAY(I)  ORB26210
    ENDIF                         ORB26220
                                ORB26230
* LAT INCREASING, LONG. DECREASING ORB26240

```

```

+ ELSEIF((ELORAY(I - 2) .GT. ELORAY(I - 1)) .AND.
+        (ELARAY(I - 2) .LE. ELARAY(I - 1))) THEN ORB26250
+ IF((ELORAY(I) .GT. 170) .OR.
+     (ELARAY(I) .LT. -80)) THEN ORB26260
+ CALL POLY3 ORB26270
+ CALL NOCHECK ORB26280
+ CALL CURVE(TLONG,TLAT,ITEMP,1) ORB26290
+ GOTO 430 ORB26300
ELSE ORB26310
    ITEMp = ITEMp + 1 ORB26320
    TLONG(ITEMP) = ELORAY(I) ORB26330
    TLAT(ITEMP) = ELARAY(I) ORB26340
ENDIF ORB26350
ORB26360
ORB26370
ORB26380
LAT. DECREASING, LONG. INCREASING ORB26390
ELSEIF((ELORAY(I - 2) .LE. ELORAY(I - 1)) .AND. ORB26400
+        (ELARAY(I - 2) .GT. ELARAY(I - 1))) THEN ORB26410
+ IF((ELORAY(I) .LT. -170) .OR. ORB26420
+     (ELARAY(I) .GT. 80)) THEN ORB26430
+ CALL POLY3 ORB26440
+ CALL NOCHECK ORB26450
+ CALL CURVE(TLONG,TLAT,ITEMP,1) ORB26460
+ GOTO 430 ORB26470
ELSE ORB26480
    ITEMp = ITEMp + 1 ORB26490
    TLONG(ITEMP) = ELORAY(I) ORB26500
    TLAT(ITEMP) = ELARAY(I) ORB26510
ENDIF ORB26520
ENDIF ORB26530
IF( I .EQ. NUM) THEN ORB26540
    CALL POLY3 ORB26550
    CALL NOCHECK ORB26560
    CALL CURVE(TLONG,TLAT,ITEMP,1) ORB26570
ENDIF ORB26580
I = I + 1 ORB26590
GOTO 440 ORB26600
ENDIF ORB26610
ORB26620
ORB26630
CALL POLY3 ORB26640
CALL NOCHECK ORB26650
CALL CURVE(TLONG,TLAT,ITEMP,1) ORB26660
ORB26670
ORB26680
CALL COMPLX ORB26690
CALL HEIGHT(.2) ORB26700
CALL THKFRM (0.03) ORB26710
CALL FRAME ORB26720
CALL RESET('COMPLX') ORB26730
CALL RESET('HEIGHT') ORB26740
CALL ENDGR (0) ORB26750
RETURN ORB26760
END ORB26770
ORB26780
ORB26790
ORB26800

```

SUBROUTINE DATA(I,A,E,TFEA,TFSU,TFMO,TFDRA,PER,PI,TDI,TDA,TDE,  
 + TDNM,TDNA,TDLAN,TDH,TDAP,MM,MA,LAN,H,AP,V,R) ORB26810  
 \* THIS SUBROUTINE Displays THE ORBITAL DATA FOR BOTH THE PERIFOCAL ORB26820  
 \* AND THE GROUND TRACK PLOTS. ORB26830  
 \* REFER TO DISSPLA USER'S MANUAL FOR EXPLANATION OF DISSPLA ORB26840  
 \* SUBROUTINES ORB26850  
 REAL I,MM,MA,LAN ORB26860  
 MU = 3.986012E+05 ORB26870  
 \* CALCULATE THE AVERAGE FORCES FROM THE TOTAL MAGNITUDE OF ORB26880  
 \* FORCE CHANGES ORB26890  
 AVGFE = TFEA/50.0 ORB26900  
 AVGFS = TFSU / 50.0 ORB26910  
 AVGFM = TFMO / 50.0 ORB26920  
 AVGFD = TFDRA / 50.0 ORB26930  
 \* CALCULATE ORBITAL ELEMENTS IN Usable UNITS ORB26940  
 PERH = PER/3600 ORB26950  
 DI = I \* (180.0/PI) ORB26960  
 DLAN = LAN \* (180.0/PI) ORB26970  
 DAP = AP \* (180.0/PI) ORB26980  
 \* CALCULATE Average CHANGE IN ELEMENTS FOR ONE PERIOD ORB26990  
 AVGDI = TDI / 50.0 ORB27000  
 AVGDA = TDA / 50.0 ORB27010  
 AVGDE = TDE / 50.0 ORB27020  
 AVGDNM = TDNM / 50.0 ORB27030  
 AVGDNA = TDNA / 50.0 ORB27040  
 AVGLAN = TDLAN / 50.0 ORB27050  
 AVGDH = TDH / 50.0 ORB27060  
 AVGDAP = TDAP / 50.0 ORB27070  
 \* CALCULATE RADIUS'S AND VELOCITIES ORB27080  
 ENR = ((V\*\*2)/2) - (MU/R) ORB27090  
 RP = A\*(1 - E) ORB27100  
 RA = A\*(1 + E) ORB27110  
 VP = SQRT(2\*(ENR + (MU/R))) ORB27120  
 VA = SQRT(2\*(ENR + (MU/RA))) ORB27130  
 \* SET DISSPLA ORB27140  
 CALL RESET(3HALL) ORB27150  
 CALL SCMPLEX ORB27160  
 CALL PHYSOR(0.0,0.0) ORB27170  
 CALL AREA2D(8.5,4.0) ORB27180  
 \* PRINT DATA ORB27190  
 CALL MESSAG('I = \$',100,0.25,3.67) ORB27200  
 CALL REALNO(DI,3,'ABUT','ABUT') ORB27210  
 CALL MESSAG(' DEG. \$',100,'ABUT','ABUT') ORB27220  
 CALL MESSAG(' A = \$',100,'ABUT','ABUT') ORB27230  
 CALL REALNO(A,1,'ABUT','ABUT') ORB27240  
 CALL MESSAG(' KMS',100,'ABUT','ABUT') ORB27250

CALL MESSAG(' E = \$',100,'ABUT','ABUT')	ORB27370
CALL REALNO(E,3,'ABUT','ABUT')	ORB27380
CALL MESSAG(' PER = \$',100,'ABUT','ABUT')	ORB27390
CALL REALNO(PERH,2,'ABUT','ABUT')	ORB27400
CALL MESSAG(' HOURS\$',100,'ABUT','ABUT')	ORB27410
CALL MESSAG('AVERAGE RATE OF CHANGE OF ELEMENTS PER SECOND \$', + 100,1.0,3.0)	ORB27420
CALL MESSAG('DI/DT = \$',100,0.25,2.67)	ORB27430
CALL REALNO(AVGDI,-2,'ABUT','ABUT')	ORB27440
CALL MESSAG(' DA/DT = \$',100,'ABUT','ABUT')	ORB27450
CALL REALNO(AVGDA,-2,'ABUT','ABUT')	ORB27460
CALL MESSAG(' DE/DT = \$',100,'ABUT','ABUT')	ORB27470
CALL REALNO(AVGDE,-2,'ABUT','ABUT')	ORB27480
CALL MESSAG('DMM/DT = \$',100,0.25,2.33)	ORB27490
CALL REALNO(AVGDM, -2,'ABUT','ABUT')	ORB27500
CALL MESSAG(' DMA/DT = \$',100,'ABUT','ABUT')	ORB27510
CALL REALNO(AVGDMA,-2,'ABUT','ABUT')	ORB27520
CALL MESSAG(' DLM/DT = \$',100,'ABUT','ABUT')	ORB27530
CALL REALNO(AVGLAN,-2,'ABUT','ABUT')	ORB27540
CALL MESSAG(' DH/DT = \$',100,0.25,2.00)	ORB27550
CALL REALNO(AVGDH,-2,'ABUT','ABUT')	ORB27560
CALL MESSAG(' DAP/DT = \$',100,'ABUT','ABUT')	ORB27570
CALL REALNO(AVGDMA,-2,'ABUT','ABUT')	ORB27580
CALL MESSAG('AVERAGE MAGNITUDE OF FORCES PER UNIT MASS (KM/S**2) +\$',100,1.0,1.67)	ORB27590
CALL MESSAG('EARTH = \$',100,0.10,1.33)	ORB27600
CALL REALNO(AVGFE,-1,'ABUT','ABUT')	ORB27610
CALL MESSAG(' MOON = \$',100,'ABUT','ABUT')	ORB27620
CALL REALNO(AVGM, -1,'ABUT','ABUT')	ORB27630
CALL MESSAG(' SUN = \$',100,'ABUT','ABUT')	ORB27640
CALL REALNO(AVGS, -1,'ABUT','ABUT')	ORB27650
CALL MESSAG(' DRAG = \$',100,'ABUT','ABUT')	ORB27660
CALL REALNO(AVGF, -1,'ABUT','ABUT')	ORB27670
CALL MESSAG('PERIGEE\$',100,2.75,1.0)	ORB27680
CALL MESSAG(' Apogee\$',100,'ABUT','ABUT')	ORB27690
CALL MESSAG('RADIUS (KM)\$',100,0.25,0.67)	ORB27700
CALL MESSAG('RP =\$',100,2.75,0.67)	ORB27710
CALL REALNO(RP,1,'ABUT','ABUT')	ORB27720
CALL MESSAG(' \$',100,'ABUT','ABUT')	ORB27730
CALL MESSAG(' RA =\$',100,'ABUT','ABUT')	ORB27740
CALL REALNO(RA,1,'ABUT','ABUT')	ORB27750
CALL MESSAG(' VELOCITY (KM/SEC)\$' 100,0.25,0.33)	ORB27760
CALL MESSAG('VP =\$',100,2.75,0.33)	ORB27770
CALL REALNO(VP,2,'ABUT','ABUT')	ORB27780
CALL MESSAG(' \$',100,'ABUT','ABUT')	ORB27790
CALL MESSAG(' VA =\$',100,'ABUT','ABUT')	ORB27800
CALL REALNO(VA,2,'ABUT','ABUT')	ORB27810
	ORB27820
	ORB27830
	ORB27840
	ORB27850
	ORB27860
	ORB27870
	ORB27880
	ORB27890
	ORB27900
	ORB27910
	ORB27920

CALL RESET('COMPLX')  
CALL ENDGR(0)  
RETURN  
END

ORB27930  
ORB27940  
ORB27950  
ORB27960  
ORB27970  
ORB27980

## APPENDIX B. COORDINATE SYSTEMS

### A. 'IJK': GEOCENTRIC - EQUATORIAL

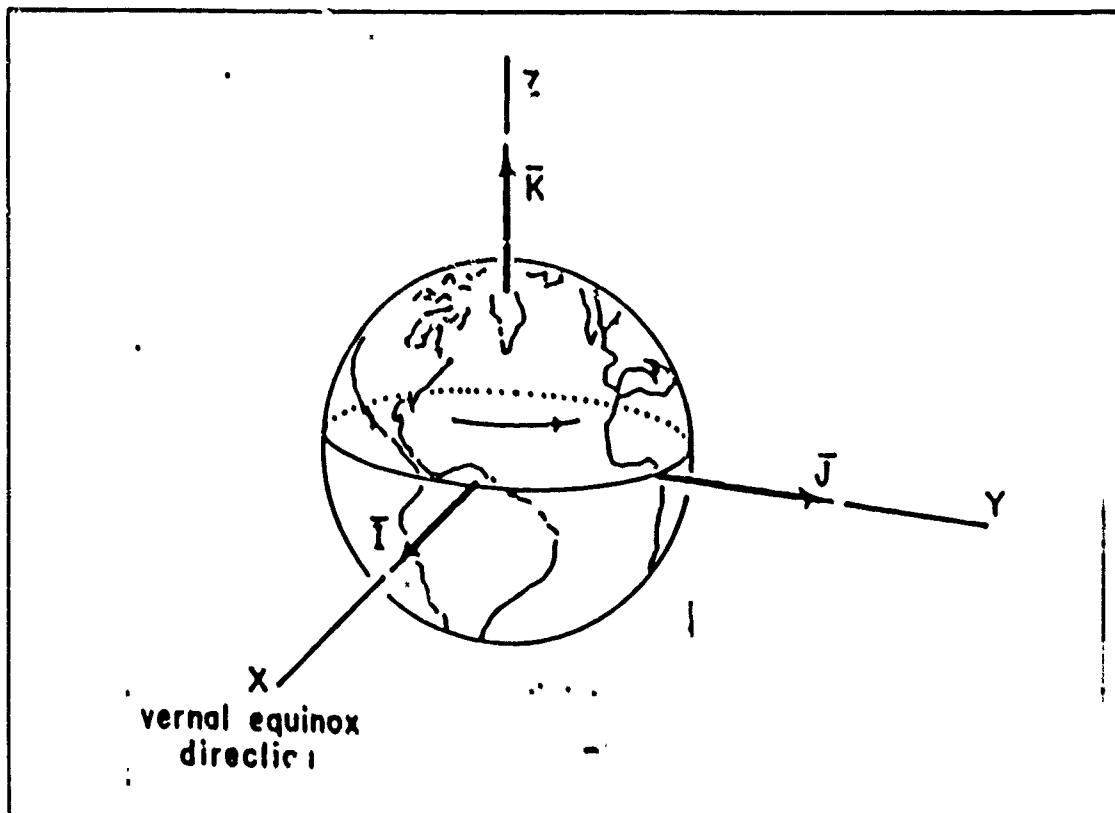


Figure 3. Geocentric-equatorial coordinate system

The geocentric-equatorial system as seen in Figure 3 has its origin at the earth's center. The fundamental plane is in the equator and the positive X-axis points in the vernal equinox direction. The Z-axis points in the direction of the north pole. This system is not fixed to the earth and turning with it; rather, the geocentric-equatorial frame is nonrotating with respect to the stars (except for precession of the equinoxes) and the earth turns relative to it. Unit vectors,  $\bar{I}$ ,  $\bar{J}$ , and  $\bar{K}$  shown in Figure 3, lie along the X, Y, and Z respectively. [Ref. 1: p.55]

## B. 'PQW': PERIFOCAL

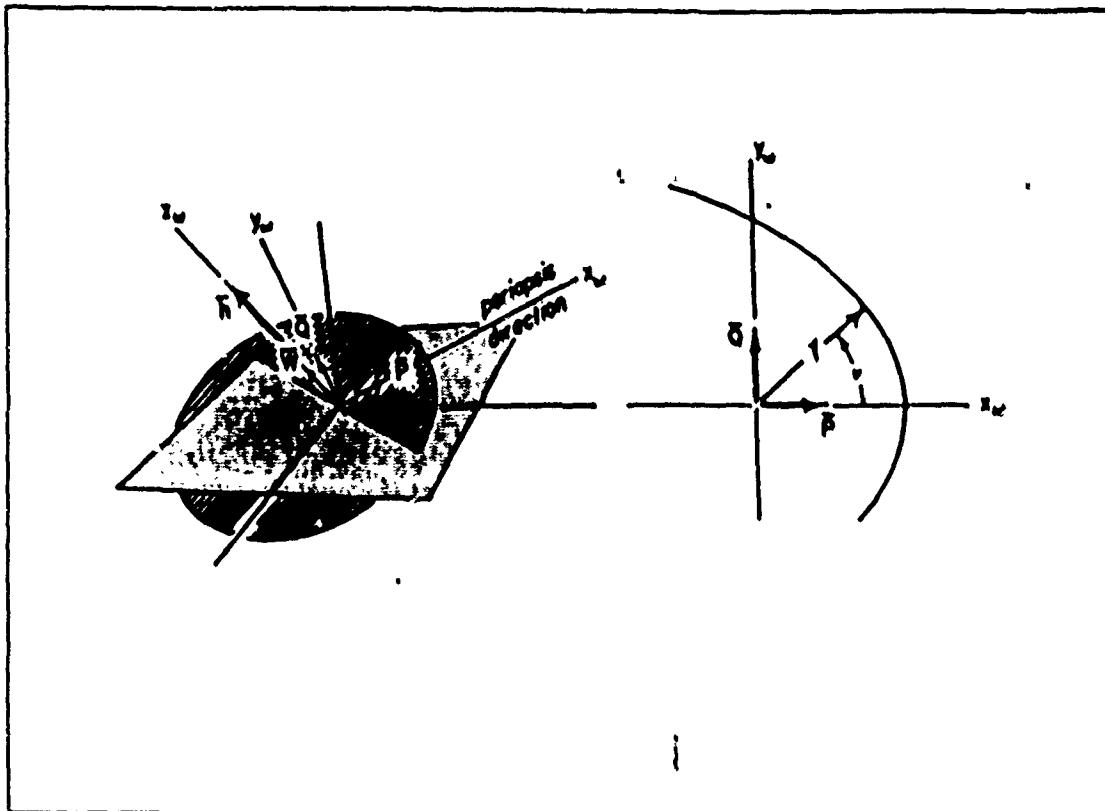


Figure 4. Perifocal coordinate system

The perifocal coordinate system has its fundamental plane in the plane of the satellite's orbit as seen in Figure 4. The coordinate axes are named,  $X_w$ ,  $Y_w$  and  $Z_w$ . The  $X_w$  axis points toward the perigee; the  $Y_w$  axis is rotated 90 degrees in the direction of orbital motion and lies in the orbital plane; the  $Z_w$  axis along  $\vec{h}$  completes the right-handed perifocal system. Unit vectors in the direction of  $X_w$ ,  $Y_w$  and  $Z_w$  are called  $\vec{P}$ ,  $\vec{Q}$  and  $\vec{W}$  respectively. [Ref. 1: p.57]

### C. 'RSW': ORBITAL

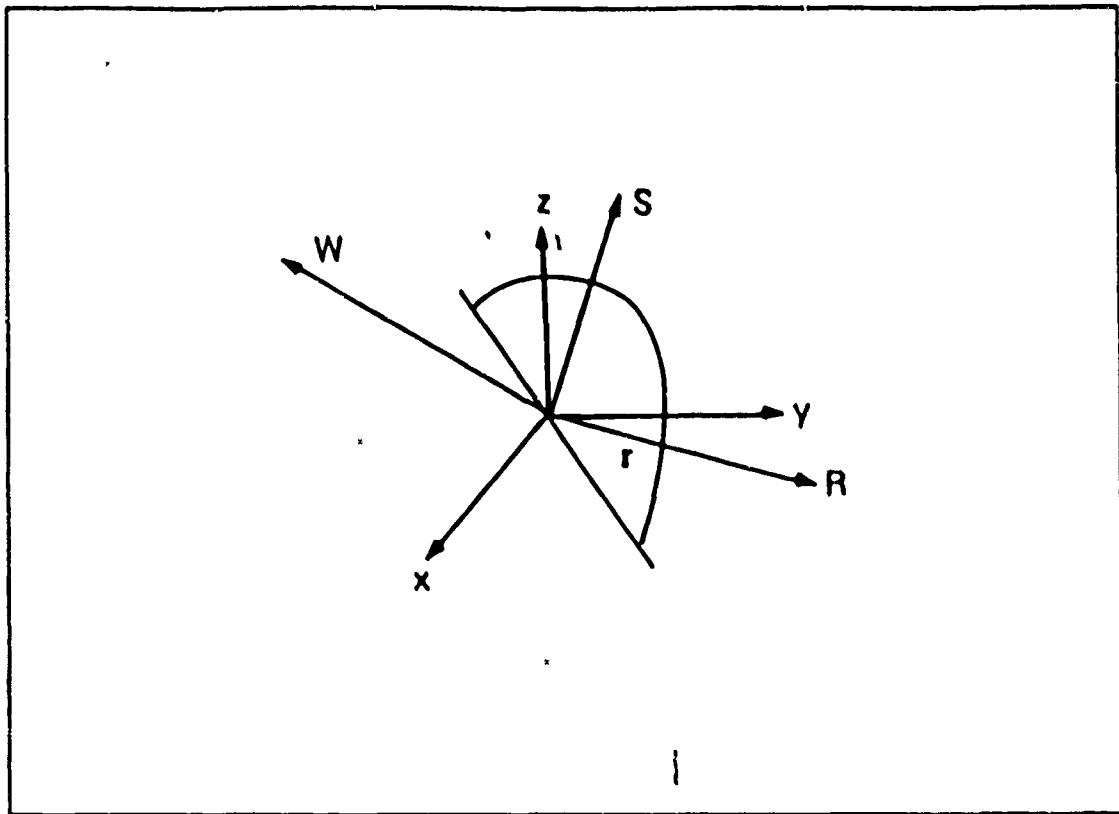


Figure 5. Orbital coordinate system

(Figure 9.4-1, Ref. 1)

The orbital coordinate system has its principle axis,  $R$  (unit vector  $r$ ), along the instantaneous radius vector,  $r$  as seen in Figure 5. The axis  $S$  is rotated 90 degrees from  $R$  in the direction of increasing true anomaly. The third axis,  $W$ , is perpendicular to both  $R$  and  $S$ . Note that this coordinate system is simply rotated  $\nu_0$  from the PQW perifocal system. [Ref. 1: p.398]

## D. COORDINATE TRANSFORMATIONS

The coordinate transformations, for the previous coordinate systems, use angular rotations about the axis to evaluate the transformation matrix. The matrix elements  $r_{ij}$  are calculated, then applied to the old vector to get the vector in the new coordinate system. The following orbital elements are used:

$\Omega$  = longitude of ascending node

$\omega$  = argument of perigee

$i$  = inclination

$u_0$  = argument of latitude

$v_0$  = true anomaly

The coordinate transformations follow [Ref. 1: p.74-83]

### 1. PQW to IJK

$$\begin{aligned}r_{11} &= \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i \\r_{12} &= -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i \\r_{13} &= \sin \Omega \cos \omega \\r_{21} &= \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i \\r_{22} &= -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i \\r_{23} &= -\cos \Omega \sin i \\r_{31} &= \sin \omega \sin i \\r_{32} &= \cos \omega \sin i \\r_{33} &= \cos i \\ \vec{I} &= r_{11}\vec{P} + r_{12}\vec{Q} + r_{13}\vec{W} \\ \vec{J} &= r_{21}\vec{P} + r_{22}\vec{Q} + r_{23}\vec{W} \\ \vec{K} &= r_{31}\vec{P} + r_{32}\vec{Q} + r_{33}\vec{W}\end{aligned}$$

### 2. IJK to PQW (inverse of #1)

$$\begin{aligned}\vec{P} &= r_{11}\vec{I} + r_{21}\vec{J} + r_{31}\vec{K} \\ \vec{Q} &= r_{12}\vec{I} + r_{22}\vec{J} + r_{32}\vec{K} \\ \vec{W} &= r_{13}\vec{I} + r_{23}\vec{J} + r_{33}\vec{K}\end{aligned}$$

### 3. IJK to RSW

$$\begin{aligned}r_{11} &= \cos \Omega \cos u_0 - \sin \Omega \sin u_0 \cos i \\r_{12} &= \sin \Omega \cos u_0 + \sin u_0 \cos \Omega \cos i \\r_{13} &= \sin i \sin u_0 \\r_{21} &= -\cos \Omega \sin u_0 - \sin \Omega \cos u_0 \cos i \\r_{22} &= -\sin \Omega \sin u_0 + \cos \Omega \cos u_0 \cos i \\r_{23} &= \cos u_0 \sin i \\r_{31} &= \sin \Omega \sin i \\r_{32} &= -\cos \Omega \sin i \\r_{33} &= \cos i \\ \vec{R} &= r_{11}\vec{I} + r_{12}\vec{J} + r_{13}\vec{K} \\ \vec{S} &= r_{21}\vec{I} + r_{22}\vec{J} + r_{23}\vec{K} \\ \vec{W} &= r_{31}\vec{I} + r_{32}\vec{J} + r_{33}\vec{K}\end{aligned}$$

4. RSW to IJK (inverse of  $\hat{\Sigma}$ )

$$\begin{aligned}\hat{I} &= r_{11}\hat{R} + r_{12}\hat{S} + r_{13}\hat{W} \\ \hat{J} &= r_{12}\hat{R} + r_{22}\hat{S} + r_{23}\hat{W} \\ \hat{K} &= r_{13}\hat{R} + r_{23}\hat{S} + r_{33}\hat{W}\end{aligned}$$

5. PQW to RSW

$$\begin{aligned}r_{11} &= \cos v_0 \\ r_{12} &= \sin v_0 \\ r_{13} &= 0.0 \\ r_{21} &= -\sin v_0 \\ r_{22} &= \cos v_0 \\ r_{23} &= 0.0 \\ r_{31} &= 0.0 \\ r_{32} &= 0.0 \\ r_{33} &= 1.0\end{aligned}$$

$$\begin{aligned}\hat{R} &= r_{11}\hat{P} + r_{12}\hat{Q} + r_{13}\hat{W} \\ \hat{S} &= r_{21}\hat{P} + r_{22}\hat{Q} + r_{23}\hat{W} \\ \hat{W} &= r_{31}\hat{P} + r_{32}\hat{Q} + r_{33}\hat{W}\end{aligned}$$

6. RSW to PQW (inverse of  $\hat{\Sigma}$ )

$$\begin{aligned}\hat{P} &= r_{11}\hat{R} + r_{21}\hat{S} + r_{31}\hat{W} \\ \hat{Q} &= r_{12}\hat{R} + r_{22}\hat{S} + r_{32}\hat{W} \\ \hat{W} &= r_{13}\hat{R} + r_{23}\hat{S} + r_{33}\hat{W}\end{aligned}$$

## APPENDIX C. ORBITAL ELEMENTS

The user is assumed to be studying orbital mechanics and should understand the orbital elements and how to calculate them. A brief description of the elements and the equations used to calculate the elements follow. For a detailed explanation of the elements and the equations to calculate them refer to Chapters 1 and 2 of reference 1. Figure 6 on page 83 shows the orbital elements in the Geocentric-Equatorial and perifocal coordinate system.

### 1. Angular Momentum ( $\vec{h}$ ):

The specific angular momentum is a constant of the motion of the satellite, defined as  $\vec{h} = \vec{r} \times \vec{v}$ .

$$\vec{h} = \vec{r} \times \vec{v} = h_i \vec{i} + h_j \vec{j} + h_k \vec{k}$$

$$h_i = r_j v_k - r_k v_j$$

$$h_j = r_k v_i - r_i v_k$$

$$h_k = r_i v_j - r_j v_i$$

$$h = \sqrt{h_i^2 + h_j^2 + h_k^2}$$

### 2. Node Vector ( $\vec{n}$ ):

The node vector is a vector pointing along the line of nodes in the direction of the ascending node.

$$\vec{n} = \vec{k} \times \vec{h} = -h_j \vec{i} + h_i \vec{j}$$

$$n = \sqrt{h_j^2 + h_i^2}$$

### 3. Semi-latus rectum ( $p$ ):

The semi-latus rectum is a geometric constant of the conic section.

$$p = \frac{h^2}{\mu}$$

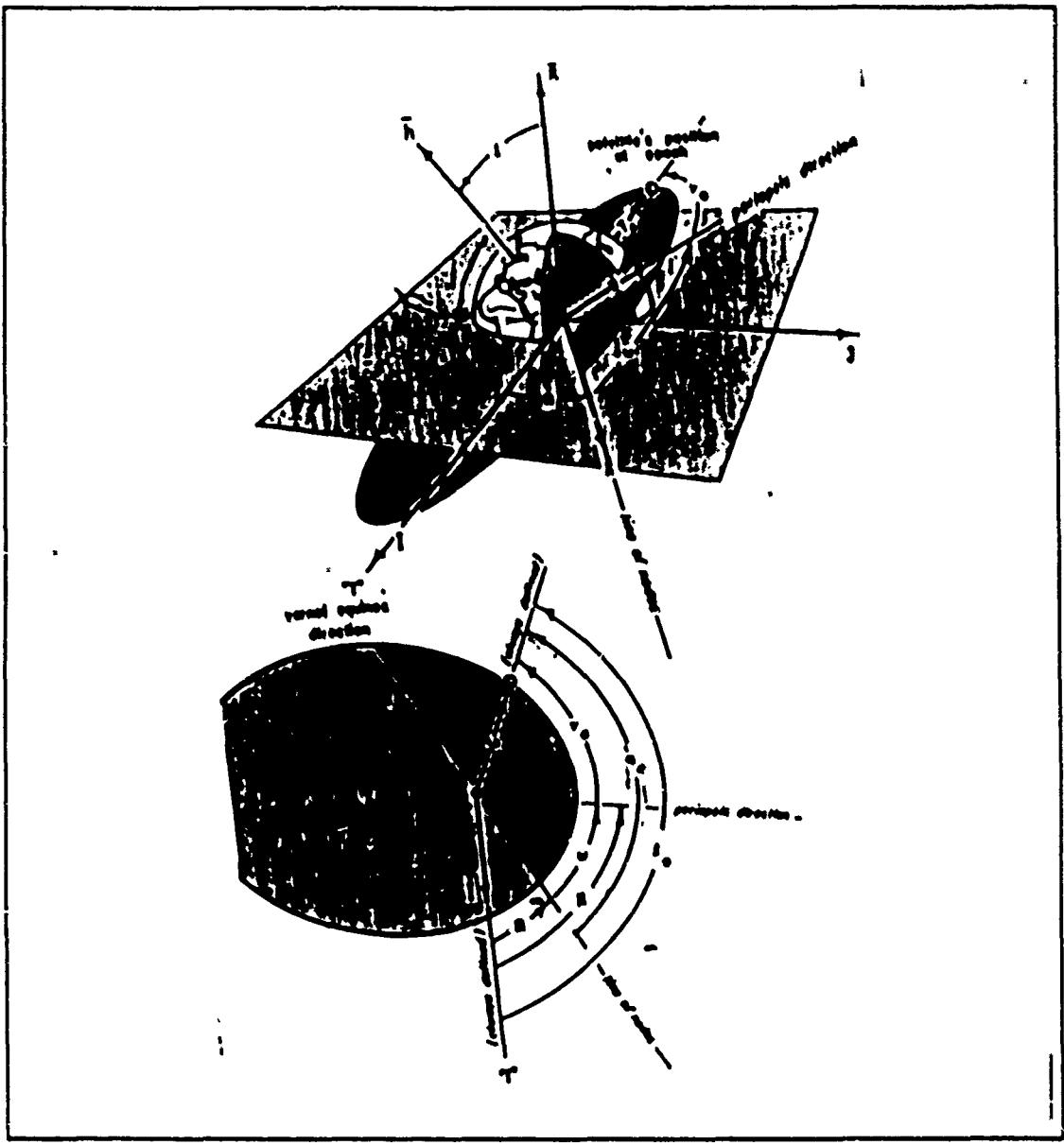
### 4. Eccentricity ( $e$ ):

The eccentricity is a constant defining the shape of the conic orbit.

$$\vec{e} = \frac{1}{\mu} \left[ \left( r^2 - \frac{\mu}{r} \right) \vec{r} - (\vec{r} \cdot \vec{r}) \vec{r} \right]$$

$$e = |\vec{e}|$$

### 5. Semi-major axis ( $a$ ):



**Figure 6. Orbital elements**

The semi-major axis is a constant defining the size of the orbit.

$$a = \frac{(1 - e^2)}{p}$$

**6. Inclination (*i*):**

The inclination is the angle between the 'K' unit vector in the 'IJK' system and the angular momentum vector, 'h'.

$$i = \cos^{-1} \left( \frac{\vec{r} \cdot \vec{k}}{h} \right) = \cos^{-1} \left( \frac{h_c}{h} \right)$$

7. Longitude of ascending node ( $\Omega$ ):

The longitude of the ascending node is the angle in the fundamental plane, between the 'I' unit vector and the point where the satellite crosses through the fundamental plane in a northerly direction (ascending node) measured counter-clockwise when viewed from the north side of the fundamental plane.

$$\Omega = \cos^{-1} \left( \frac{n_c}{n} \right)$$

8. Argument of perigee ( $\omega$ ):

The argument of perigee is the angle in the plane of the satellite's orbit, between the ascending node and the perigee point, measured in the direction of the satellite's motion.

$$\omega = \cos^{-1} \left( \frac{\vec{n} \cdot \vec{e}_l}{n e_l} \right) = \cos^{-1} \frac{(n_l c_l + n_c e_l)}{n e_l}$$

9. True anomaly at epoch ( $v_0$ ):

The true anomaly at epoch is the angle in the plane of the satellite's orbit, between perigee and the position of the satellite at a particular time,  $t_0$ , called the "epoch".

$$v_0 = \cos^{-1} \left( \frac{\vec{e}_l \cdot \vec{r}}{e_l r} \right)$$

10. Argument of latitude ( $u$ ):

The argument of latitude is the angle in the plane of the orbit, between the ascending node and the radius vector to the satellite at time  $t_0$ .

$$u_0 = \cos^{-1} \left( \frac{\vec{n} \cdot \vec{r}}{n r} \right)$$

11. Longitude of perigee ( $\Pi$ ):

The longitude of perigee is the angle from 'I' to perigee measured eastward to the ascending node and then in the orbital plane to perigee.

$$\Pi = \Omega + \omega$$

12. True longitude at epoch ( $l_0$ ):

The true longitude at epoch is the angle between 'I' and  $r_0$  (the radius vector to the satellite at  $t_0$ ) measured eastward to the ascending node and then in the orbital plane to  $r_0$ .

$$l_0 = \omega + \Omega + v_0$$

13. Period (per):

The period is the time the for the satellite to complete one orbit.

$$Per = 2 \sqrt{\frac{a^3}{\mu}}$$

14. Eccentric anomaly (EA):

The eccentric anomaly is the angle between the perigee and a position on an auxiliary circle circumscribed about the ellipse where a perpendicular line to the major axis has been extended from the epoch location of the satellite to the auxiliary circle.

$$E.A = \cos^{-1} \frac{e + \cos(v)}{1 + e \cos(v)}$$

15. Mean motion ( $n'$ ):

The mean motion is defined below:

$$n' = \sqrt{\frac{\mu}{a^3}}$$

16. Mean anomaly (MA):

The mean anomaly is defined below:

$$M.A = n'(t - T) = E.A - e \sin(E.A)$$

17. Time of flight (TOF):

The time of flight is the elapsed time from when the satellite was at perigee to the current epoch.

$$(t - T) = \sqrt{\frac{a^3}{\mu}} (E.A - e \sin(E.A))$$

## APPENDIX D. SAMPLE ORBITS

To demonstrate the capabilities of the program, a variety of orbital plots will follow:

### I. Low earth orbit (LEO).

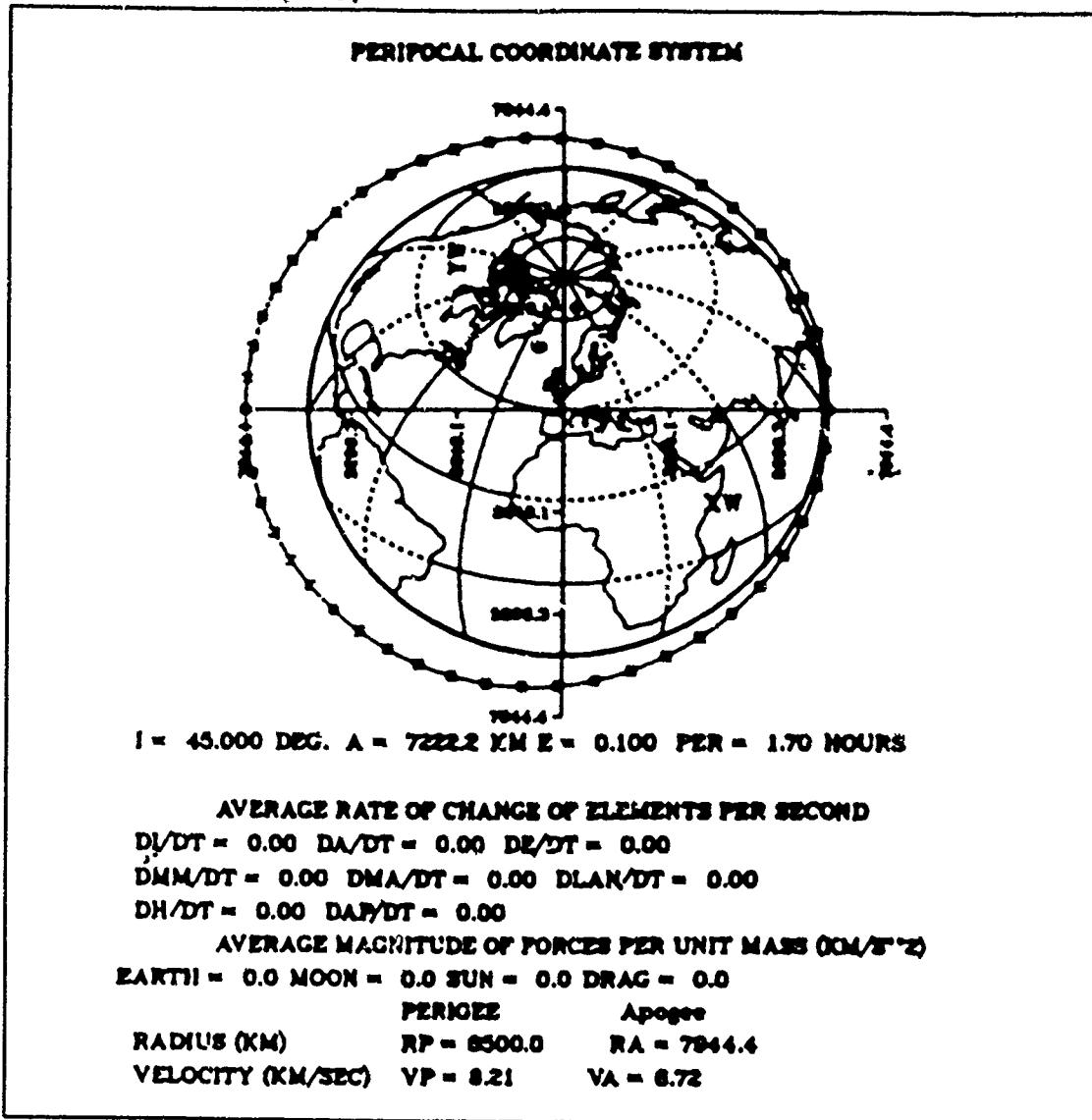


Figure 7. Unperturbed Low Earth Orbit (LEO)

Figure 7 shows the perifocal plot of a satellite in an unperturbed low earth orbit (LEO). The initial parameters of the orbit were entered as follows:  
 radius of perigee (RP) = 6500 km  
 eccentricity (e) = 0.1  
 inclination (i) = 45 degrees.

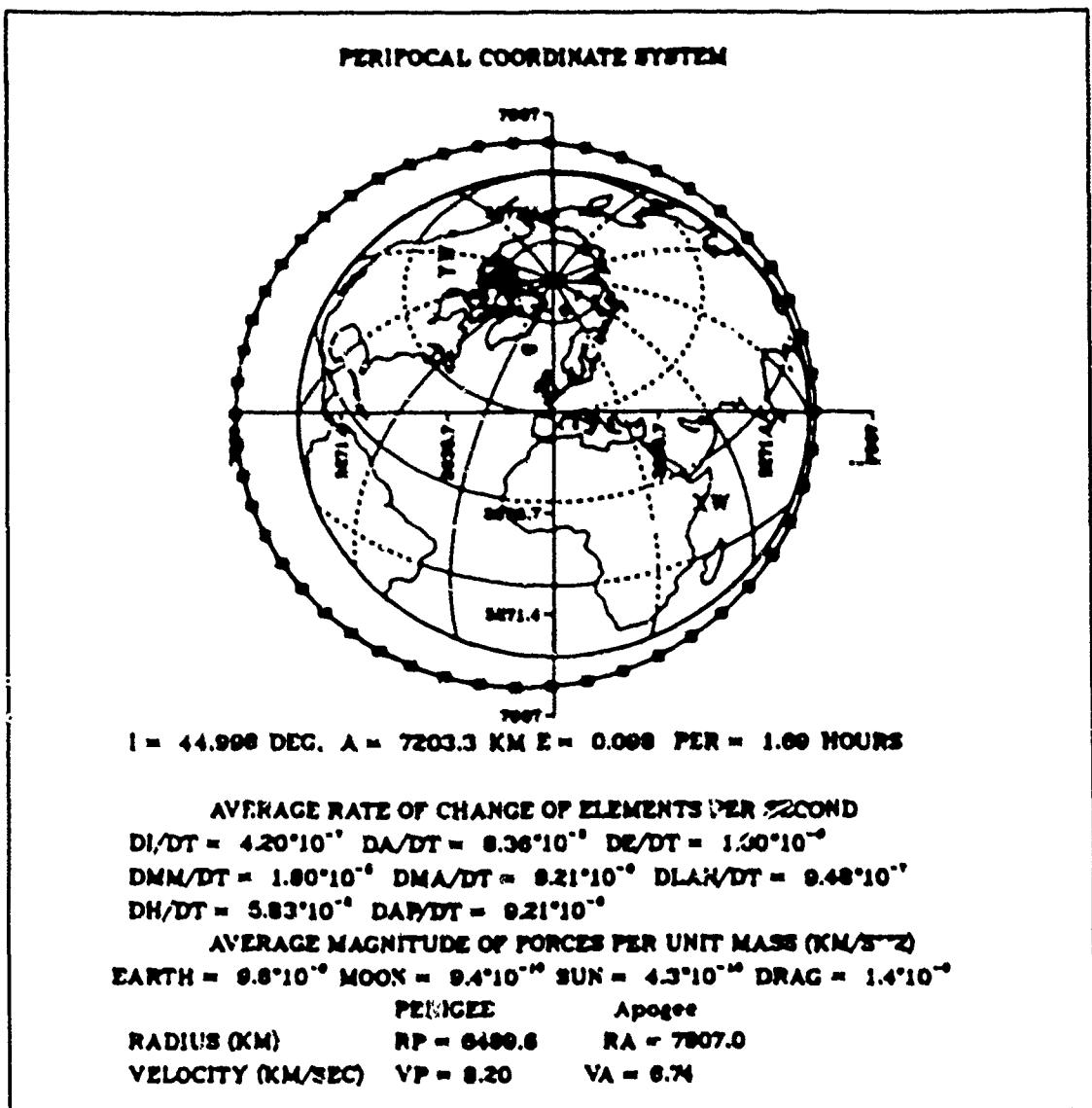


Figure 8. Perturbed Low Earth Orbit (LEO)

With perturbing forces applied to the previous LEO, the drag force will be the dominate perturbing force. The drag will act as a negative velocity change applied in the area of perigee, with the result of decreasing the semi-major axis length, this in effect will decrease the eccentricity of the orbit, as can be seen by comparing the orbital data of the unperturbed LEO in Figure 7 on page S6 with the orbital data of the perturbed LEO in Figure 8.

2. Circular orbit.

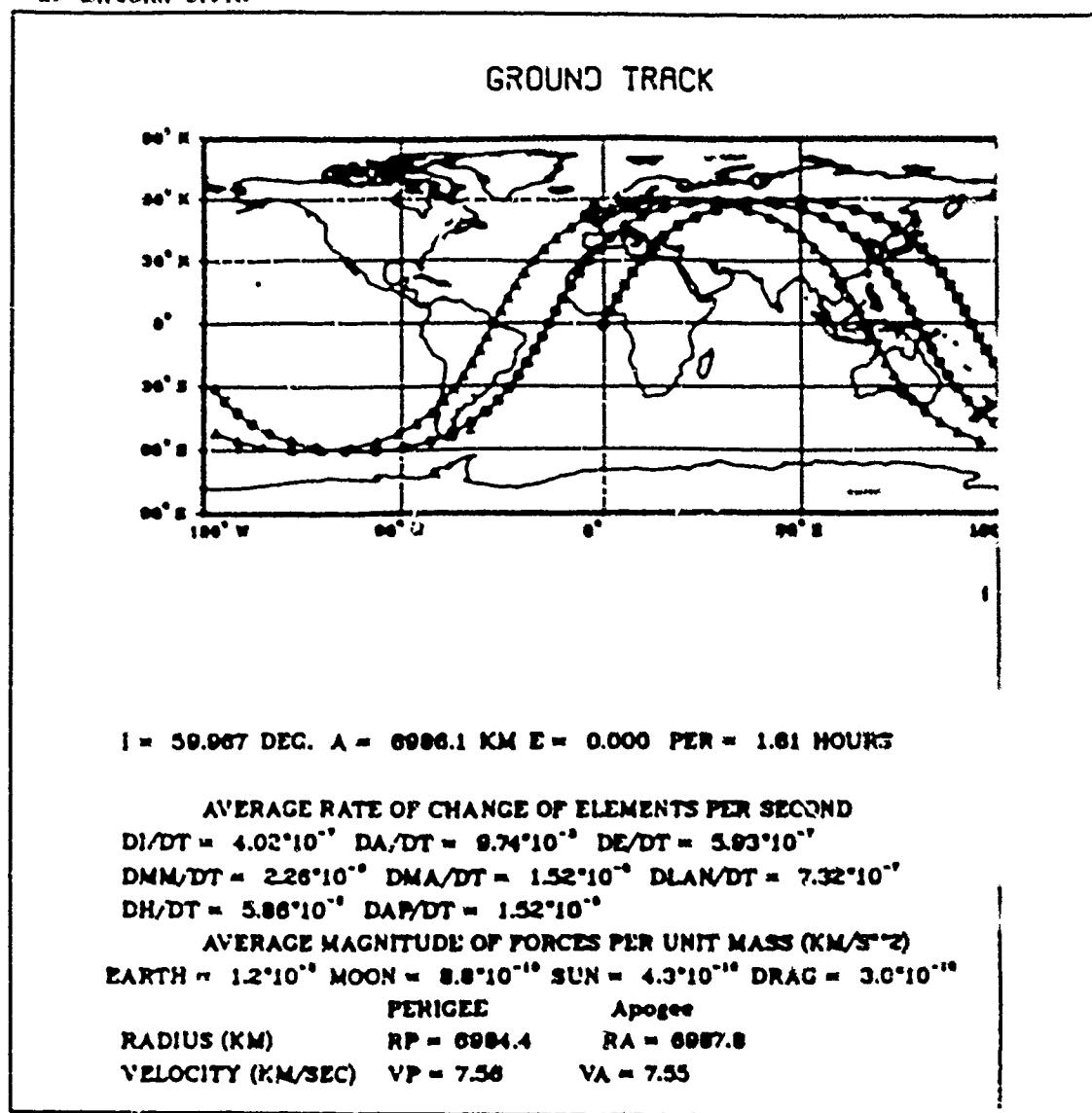


Figure 9. Circular Orbit

An example of the plot of the ground track of a sequence of three 60 degree inclined perurbed circular orbits with a radius of 7000 km is shown in Figure 9. The sequence of orbits displays the precession of the orbit around the earth.

3. Transfer orbit.

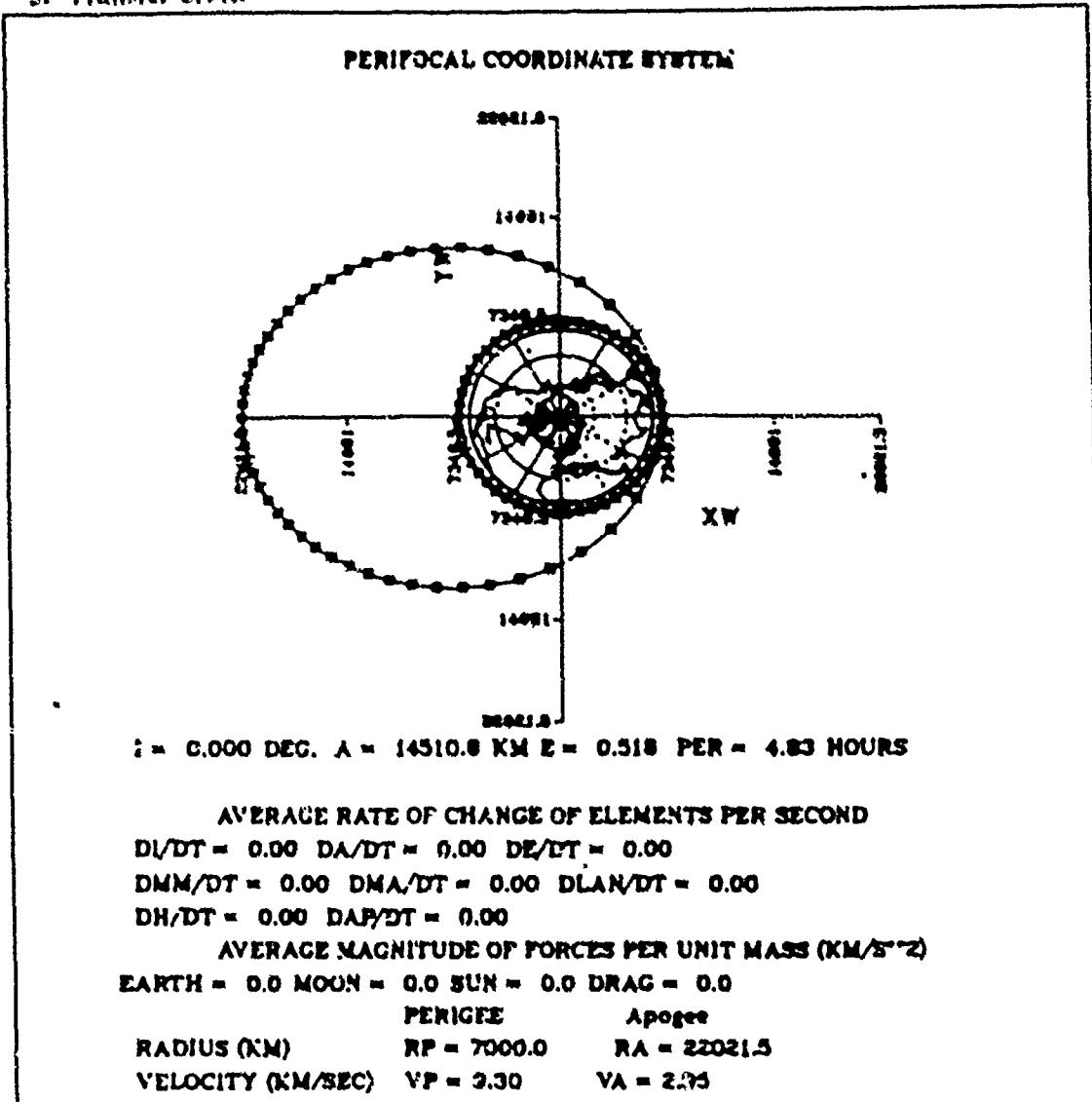


Figure 10. Transfer Orbit

The transfer orbit between a circular, equatorial LEO and a molniya orbit (high eccentric orbit) is shown in Figure 10. A velocity increase of 1.75 km/s was applied at the perigee to simulate a perigee kick to boost the satellite into the molniya orbit. A similar velocity change could then be applied at apogee to create a high altitude circular orbit, or a negative velocity change applied at perigee could be used to bring the satellite back to a LEO.

#### 4. Geosynchronous orbit

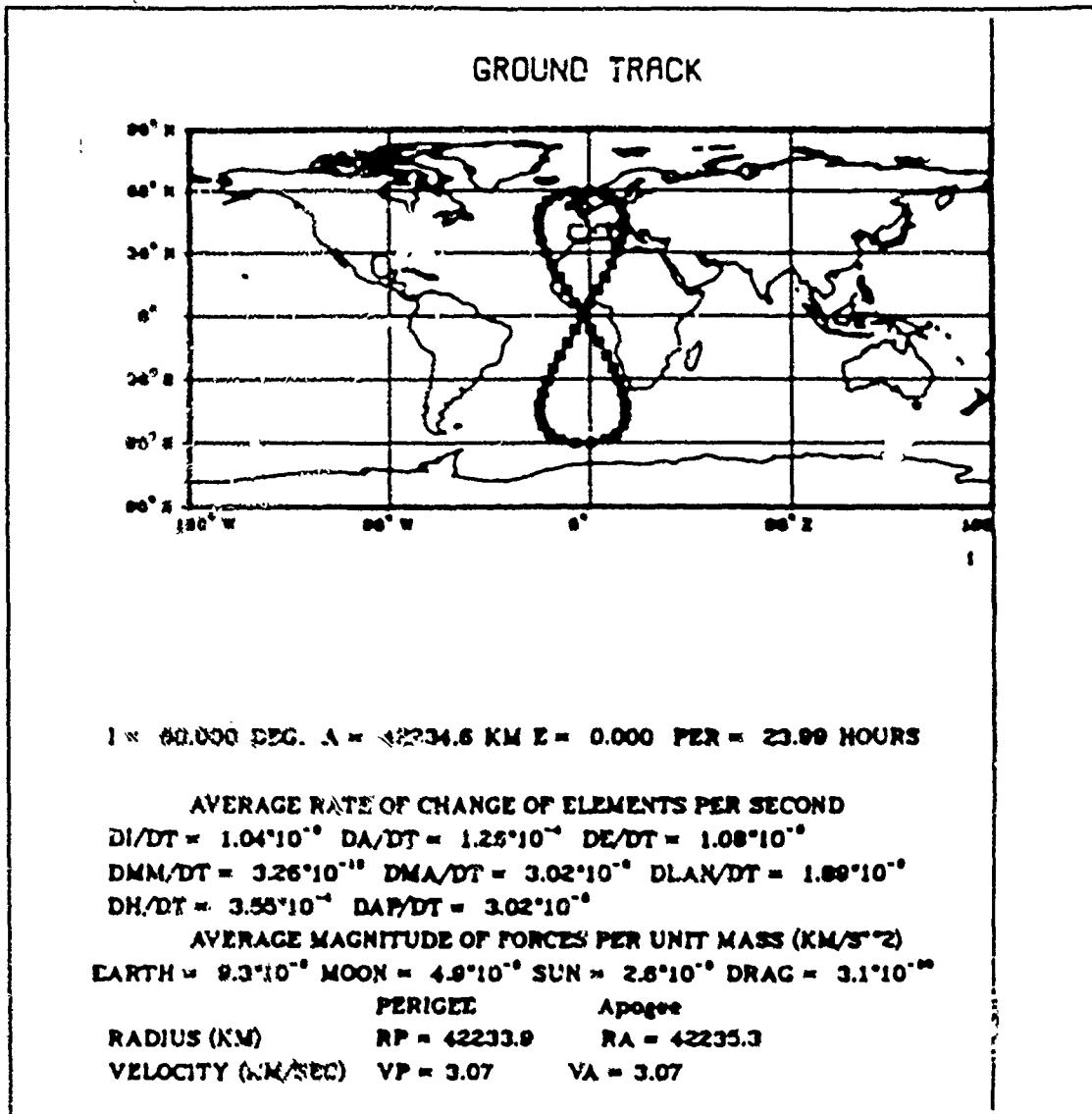


Figure 11. Geosynchronous Orbit

The ground track of a perturbed geosynchronous orbit inclined 60 degrees is shown in Figure 11. The orbit displays the figure eight typical with inclined geosynchronous orbits.

## LIST OF REFERENCES

1. Bate, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Dover Publications, Inc., 1971.
2. Martin Marietta Corporation, Space Systems Division, *Orbital Flight Handbook*, volume 1, 1963.
3. Agrawal, B.N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., 1968.
4. Roy, A.E., *Orbital Motion*, Adam Hilger Ltd., 1982.
5. Computer Associates, *DISSPLA User's Manual*, version 10.0, 1987.

## INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Commander Naval Space Command Attn: Code N3 Dahlgren, VA 22448	1
4. Commander United States Space Command Attn: Technical Library Peterson AFB, CO 80914	1
5. Director Navy Space System Division Chief of Naval Operations (OP-943) Washington, D.C. 20305-2000	1
6. Superintendent Naval Postgraduate School Space System Academic Group Attn: Prof. R. Panholzer (Code 62Pz) Monterey, CA 93943-5000	1
7. Superintendent Naval Postgraduate School Attn: Prof. E. Milne (Code 61Mn) Monterey, CA 93943-5000	1
8. Superintendent Naval Postgraduate School Attn: Prof. D. Danielson (Code 53Dd) Monterey, CA 93943-5000	1
9. Commanding Officer Attn: LT Anthony Vraa Department Head School Class 111 Surface Warfare Officer School Newport, RI 02841-5012	2